
Civis Client Documentation

Release 1.0.0

Civis Analytics

Feb 01, 2018

Contents

1	Installation	3
2	Python version support	5
3	Authentication	7
4	User Guide	9
5	Client API Reference	11
6	Indices and tables	677
	Python Module Index	679

The Civis API Python client is a Python package that helps analysts and developers interact with the Civis Platform. The package includes a set of tools around common workflows as well as a convenient interface to make requests directly to the Civis API.

CHAPTER 1

Installation

The recommended install method is `pip`:

```
pip install civis
```

Alternatively, you may clone the code from github and build from source:

```
git clone https://github.com/civisanalytics/civis-python.git
cd civis-python
python setup.py install
```

The client has a soft dependency on `pandas` to support features such as data type parsing. If you are using the `io` namespace to read or write data from Civis, it is highly recommended that you install `pandas` and set `use_pandas=True` in functions that accept that parameter. To install `pandas`:

```
pip install pandas
```

Machine learning features in the `ml` namespace have a soft dependency on `scikit-learn` and `pandas`. Install `scikit-learn` to export your trained models from the Civis Platform or to provide your own custom models. Use `pandas` to download model predictions from the Civis Platform. The `civis.ml` code optionally uses the `feather` format to transfer data from your local computer to Civis Platform. Install these dependencies with

```
pip install scikit-learn
pip install pandas
pip install feather-format
```

Some CivisML models have open-source dependencies in addition to `scikit-learn`, which you may need if you want to download the model object. These dependencies are `civismml-extensions`, `glmnet`, and `muffnn`. Install these dependencies with

```
pip install civismml-extensions
pip install glmnet
pip install muffnn
```


CHAPTER 2

Python version support

Python 2.7, 3.4, 3.5, and 3.6

CHAPTER 3

Authentication

In order to make requests to the Civis API, you will need an API key that is unique to you. Instructions for creating a new key are found here: <https://civis.zendesk.com/hc/en-us/articles/216341583-Generating-an-API-Key>. By default, the Python client will look for your key in the environment variable `CIVIS_API_KEY`. To add the API key to your environment, copy the key you generated to your clipboard and follow the instructions below for your operating system.

Mac

Open `.bash_profile` in TextEdit:

```
cd ~/
touch .bash_profile
open -e .bash_profile
```

Then add the following line, replacing `api_key_here` with your key:

```
export CIVIS_API_KEY="api_key_here"
```

Linux

Open `.bash_profile` in your favorite editor (nano is used here):

```
cd ~/
nano .bash_profile
```

Then add the following line, replacing `api_key_here` with your key:

```
export CIVIS_API_KEY="api_key_here"
```


CHAPTER 4

User Guide

For a more detailed walkthrough, see the *User Guide*.

5.1 User Guide

5.1.1 Getting Started

After installing the Civis API Python client and setting up your API key, you can now import the package `civis`:

```
>>> import civis
```

There are two entrypoints for working with the Civis API. The first is the `civis` namespace, which contains tools for typical workflows in a user friendly manner. For example, you may want to perform some transformation on your data in Python that might be tricky to code in SQL. This code downloads data from Civis, calculates the correlation between all the columns and then uploads the data back into Civis:

```
>>> df = civis.io.read_civis(table="my_schema.my_table",
...                         database="database",
...                         use_pandas=True)
>>> correlation_matrix = df.corr()
>>> correlation_matrix["corr_var"] = correlation_matrix.index
>>> fut = civis.io.dataframe_to_civis(df=correlation_matrix,
...                                  database="database",
...                                  table="my_schema.my_correlations")
...
>>> fut.result()
```

5.1.2 Civis Futures

In the code above, `dataframe_to_civis()` returns a special `CivisFuture` object. Making a request to the Civis API usually results in a long running job. To account for this, various functions in the `civis` namespace return a `CivisFuture` to allow you to process multiple long running jobs simultaneously. For instance, you may want to start many jobs in parallel and wait for them all to finish rather than wait for each job to finish before starting the next one.

The `CivisFuture` follows the `concurrent.futures.Future` API fairly closely. For example, calling `result()` on `future` above forces the program to wait for the job started with `dataframe_to_civis()` to finish and returns the result.

You can create `CivisFuture` objects for many tasks (e.g., scripts, imports). Here, we will create a container script that does the simple task of printing the text “HELLO WORLD”, execute it, and then wait for it to finish.

```
>>> import civis
>>> import concurrent.futures
>>>
>>> client = civis.APIClient()
>>>
>>> # Create a container script. This is just a simple example. Futures can
>>> # also be used with SQL queries, imports, etc.
>>> response_script = client.scripts.post_containers(
...     required_resources={'cpu': 512, 'memory': 1024},
...     docker_command="echo 'HELLO WORLD'",
...     docker_image_name='civisanalytics/datascience-python')
>>> script_id = response_script.id
>>>
>>> # Create a run in order to execute the script.
>>> response_run = client.scripts.post_containers_runs(script_id)
>>> run_id = response_run.id
>>>
>>> # Create a future to represent the result of the run.
>>> future = civis.futures.CivisFuture(
...     client.scripts.get_containers_runs, (script_id, run_id))
>>>
>>> # You can then have your code block and wait for the future to be done as
>>> # follows.
>>> concurrent.futures.wait([future])
>>>
>>> # Alternatively, you can call `future.result()` to block and get the
>>> # status of the run once it finishes. If the run is already completed, the
>>> # result will be returned immediately.
>>> result = future.result()
>>>
>>> # Alternatively, one can start a run and get a future for it with the helper
>>> # function `civis.utils.run_job`:
>>> future2 = civis.utils.run_job(script_id)
>>> future2.result()
```

5.1.3 Working Directly with the Client

Although many common workflows are included in the Civis API Python client, projects often require direct calls to the Civis API. For convenience, the Civis API Python client implements an `APIClient` object to make these API calls with Python syntax rather than a manually crafted HTTP request. To make a call, first instantiate an `APIClient` object:

```
>>> client = civis.APIClient()
```

Note: Creating an instance of `APIClient` makes an HTTP request to determine the functions to attach to the object. You must have an API key and internet connection to create an `APIClient` object. By default, the functions attached to the object come from a base set of Civis API endpoints. Based on your user profile, you may have access to a set of developmental endpoints. To access these, instantiate the client with `client = civis.`

```
APIClient(resources='all').
```

With the client object instantiated, you can now make API requests like listing your user information:

```
>>> client.users.list_me()
{'email': 'user@email.com',
 'feature_flags': {'left_nav_basic': True,
                   'results': True,
                   'scripts_notify': True,
                   'table_person_matching': True},
 'id': 1,
 'initials': 'UN',
 'name': 'User Name',
 'username': 'uname'}
```

Suppose we did not have the `civis.io` namespace. This is how we might export a CSV file from Civis. As you will see, this can be quite involved and the `civis` namespace entrypoint should be preferred whenever possible.

First, we get the ID for our database then we get the default credential for the current user.

```
>>> db_id = client.get_database_id('cluster-name')
>>> cred_id = client.default_credential
```

In order to export a table, we need to write some SQL that will generate the data to export. Then we create the export job and run it.

```
>>> generate_table = "select * from schema.tablename"
>>> export_job = client.scripts.post_sql(name="our export job",
                                         remote_host_id=db_id,
                                         credential_id=cred_id,
                                         sql=generate_table)
>>> export_run = client.scripts.post_sql_runs(export_job.id)
```

We can then poll and wait for the export to be completed.

```
>>> import time
>>> export_state = client.scripts.get_sql_runs(export_job.id,
                                              export_run.id)
...
>>> while export_state.state in ['queued', 'running']:
...     time.sleep(60)
...     export_state = client.scripts.get_sql_runs(export_job.id,
                                              export_run.id)
... 
```

Now, we can get the URL of the exported csv. First, we grab the result of our export job.

```
>>> export_result = client.scripts.get_sql_runs(export_job.id,
                                              export_run.id)
... 
```

In the future, a script may export multiple jobs, so the output of this is a list.

The path returned will have a gzipped csv file, which we could load, for example, with `pandas`.

```
>>> url = export_result.output[0].path
```

5.2 Data Import and Export

The `civis.io` namespace provides several functions for moving data in and out of Civis.

5.2.1 Tables

Often, your data will be in structured format like a table in a relational database, a CSV or a dataframe. The following functions handle moving structured data to and from Civis. When using these functions, it is recommended to have *pandas* installed and to pass `use_pandas=True` in the appropriate functions. If *pandas* is not installed, data returned from Civis will all be treated as strings.

<code>civis_to_csv(filename, sql, database[, ...])</code>	Export data from Civis to a local CSV file.
<code>civis_to_multifile_csv(sql, database[, ...])</code>	Unload the result of SQL query and return presigned urls.
<code>csv_to_civis(filename, database, table[, ...])</code>	Upload the contents of a local CSV file to Civis.
<code>dataframe_to_civis(df, database, table[, ...])</code>	Upload a <i>pandas DataFrame</i> into a Civis table.
<code>read_civis(table, database[, columns, ...])</code>	Read data from a Civis table.
<code>read_civis_sql(sql, database[, use_pandas, ...])</code>	Read data from Civis using a custom SQL string.

`civis.io.civis_to_csv`

`civis.io.civis_to_csv(filename, sql, database, job_name=None, api_key=None, client=None, credential_id=None, include_header=True, compression='none', delimiter=',', unquoted=False, archive=False, hidden=True, polling_interval=None)`

Export data from Civis to a local CSV file.

The custom SQL string will be executed twice; once to attempt to retrieve headers and once to retrieve the data. This is done to use a more performant method for retrieving the data. The first execution of the custom SQL is controlled such that changes in state cannot occur (e.g., INSERT, UPDATE, DELETE, etc.).

Parameters `filename` : str

Download exported data into this file.

`sql` : str, optional

The SQL select string to be executed.

`database` : str or int

Export data from this database. Can be the database name or ID.

`job_name` : str, optional

A name to give the job. If omitted, a random job name will be used.

`api_key` : DEPRECATED str, optional

Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

`client` : `civis.APIClient`, optional

If not provided, an `civis.APIClient` object will be created from the `CIVIS_API_KEY`.

`credential_id` : str or int, optional

The ID of the database credential. If `None`, the default credential will be used.

include_header: bool, optional

If `True`, the first line of the CSV will be headers. Default: `True`.

compression: str, optional

Type of compression to use, if any. One of `'none'`, `'zip'`, or `'gzip'`. Default `'none'`. `'gzip'` currently returns a file with no compression unless `include_header` is set to `False`. In a future release, a `'gzip'` compressed file will be returned for all cases.

delimiter: str, optional

Which delimiter to use, if any. One of `' '`, `' '`, or `'|'`. Default: `' '`.

unquoted: bool, optional

Whether or not to quote fields. Default: `False`.

polling_interval : int or float, optional

Number of seconds to wait between checks for query completion.

archive : bool, optional (deprecated)

If `True`, archive the import job as soon as it completes.

hidden : bool, optional

If `True` (the default), this job will not appear in the Civis UI.

Returns results : *CivisFuture*

A *CivisFuture* object.

See also:

`civis.io.read_civis` Read table contents into memory.

`civis.io.read_civis_sql` Read results of a SQL query into memory.

Examples

```
>>> sql = "SELECT * FROM schema.table"
>>> fut = civis_to_csv("file.csv", sql, "my_database")
>>> fut.result() # Wait for job to complete
```

civis.io.civis_to_multifile_csv

`civis.io.civis_to_multifile_csv(sql, database, job_name=None, api_key=None, client=None, credential_id=None, include_header=True, compression='none', delimiter='|', unquoted=False, prefix=None, polling_interval=None, hidden=True)`

Unload the result of SQL query and return presigned urls.

This function is intended for unloading large queries/tables from redshift as it uses a 'PARALLEL ON' S3 unload. It returns a similar manifest file to conventional S3 UNLOAD statements except the CSV parts are accessible via both files endpoint IDs and presigned S3 urls.

Parameters sql : str, optional

The SQL select string to be executed.

database : str or int

Execute the query against this database. Can be the database name or ID.

job_name : str, optional

A name to give the job. If omitted, a random job name will be used.

api_key : DEPRECATED str, optional

Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

client : `civis.APIClient`, optional

If not provided, an `civis.APIClient` object will be created from the `CIVIS_API_KEY`.

credential_id : str or int, optional

The database credential ID. If `None`, the default credential will be used.

include_header: bool, optional

If `True` include a key in the returned dictionary containing a list of column names. Default: `True`.

compression: str, optional

Type of compression to use, if any. One of `'none'`, `'zip'`, or `'gzip'`. Default `'none'`.

delimiter: str, optional

Which delimiter to use, if any. One of `' '`, `' '`, or `'|'`. Default: `'|'`.

unquoted: bool, optional

Whether or not to quote fields. Default: `False`.

prefix: str, optional

A user specified filename prefix for the output file to have. Default: `None`.

polling_interval : int or float, optional

Number of seconds to wait between checks for query completion.

hidden : bool, optional

If `True` (the default), this job will not appear in the Civis UI.

Returns `unload_manifest`: dict

A dictionary resembling an AWS manifest file. Has the following keys:

‘query’: str The query.

‘header’: list of str The columns from the query.

‘entries’: list of dict Each dict has the following keys:

‘id’: int File ID

‘name’: str Filename

‘size’: int File size in bytes

‘url’: str Unsigned S3 URL (`‘s3://...’`)

‘url_signed’: str Signed S3 URL (‘https://...’)
‘unquoted’: bool Whether the cells are quoted.
‘compression’: str Type of compression used.
‘delimiter’: str Delimiter that separates the cells.

See also:

`civis.APIClient.scripts.post_sql`

Examples

```
>>> sql = "SELECT * FROM schema.my_big_table"
>>> database = "my_database"
>>> delimiter = "|"
>>> manifest = civis_to_multifile_csv(sql, database, delimiter=delimiter)
>>> ids = [entry['id'] for entry in manifest['entries']]
>>> buf = BytesIO()
>>> civis_to_file(ids[0], buf)
>>> buf.seek(0)
>>> df = pd.read_csv(buf, delimiter=delimiter)
```

civis.io.csv_to_civis

`civis.io.csv_to_civis` (filename, database, table, api_key=None, client=None, max_errors=None, existing_table_rows='fail', diststyle=None, distkey=None, sortkey1=None, sortkey2=None, delimiter=',', headers=None, credential_id=None, polling_interval=None, archive=False, hidden=True)

Upload the contents of a local CSV file to Civis.

Parameters filename : str

Upload the contents of this file.

database : str or int

Upload data into this database. Can be the database name or ID.

table : str

The schema and table you want to upload to. E.g., 'scratch.table'.

api_key : DEPRECATED str, optional

Your Civis API key. If not given, the CIVIS_API_KEY environment variable will be used.

client : `civis.APIClient`, optional

If not provided, an `civis.APIClient` object will be created from the CIVIS_API_KEY.

max_errors : int, optional

The maximum number of rows with errors to remove from the import before failing.

existing_table_rows : str, optional

The behaviour if a table with the requested name already exists. One of 'fail', 'truncate', 'append' or 'drop'. Defaults to 'fail'.

diststyle : str, optional

The distribution style for the table. One of 'even', 'all' or 'key'.

distkey : str, optional

The column to use as the distkey for the table.

sortkey1 : str, optional

The column to use as the sortkey for the table.

sortkey2 : str, optional

The second column in a compound sortkey for the table.

delimiter : string, optional

The column delimiter. One of ',', '\t' or '|'.

headers : bool, optional

Whether or not the first row of the file should be treated as headers. The default, `None`, attempts to autodetect whether or not the first row contains headers.

credential_id : str or int, optional

The ID of the database credential. If `None`, the default credential will be used.

polling_interval : int or float, optional

Number of seconds to wait between checks for job completion.

archive : bool, optional (deprecated)

If `True`, archive the import job as soon as it completes.

hidden : bool, optional

If `True` (the default), this job will not appear in the Civis UI.

Returns results : *CivisFuture*

A *CivisFuture* object.

Notes

This reads the contents of *filename* into memory.

Examples

```
>>> with open('input_file.csv', 'w') as _input:
...     _input.write('a,b,c\n1,2,3')
>>> fut = civis.io.csv_to_civis('input_file.csv',
...                             'my-database',
...                             'scratch.my_data')
>>> fut.result()
```

civis.io.dataframe_to_civis

```
civis.io.dataframe_to_civis(df, database, table, api_key=None, client=None, max_errors=None,
                             existing_table_rows='fail', diststyle=None, distkey=None,
                             sortkey1=None, sortkey2=None, headers=None, credential_id=None,
                             polling_interval=None, archive=False, hidden=True,
                             **kwargs)
```

Upload a *pandas DataFrame* into a Civis table.

The *DataFrame*'s index will not be included. To store the index along with the other values, use *df.reset_index()* instead of *df* as the first argument to this function.

Parameters *df*: `pandas.DataFrame`

The *DataFrame* to upload to Civis.

database: str or int

Upload data into this database. Can be the database name or ID.

table: str

The schema and table you want to upload to. E.g., 'scratch.table'.

api_key: DEPRECATED str, optional

Your Civis API key. If not given, the CIVIS_API_KEY environment variable will be used.

client: `civis.APIClient`, optional

If not provided, an `civis.APIClient` object will be created from the CIVIS_API_KEY.

max_errors: int, optional

The maximum number of rows with errors to remove from the import before failing.

existing_table_rows: str, optional

The behaviour if a table with the requested name already exists. One of 'fail', 'truncate', 'append' or 'drop'. Defaults to 'fail'.

diststyle: str, optional

The distribution style for the table. One of 'even', 'all' or 'key'.

distkey: str, optional

The column to use as the distkey for the table.

sortkey1: str, optional

The column to use as the sortkey for the table.

sortkey2: str, optional

The second column in a compound sortkey for the table.

headers: bool, optional

Whether or not the first row of the file should be treated as headers. The default, `None`, attempts to autodetect whether or not the first row contains headers.

credential_id: str or int, optional

The ID of the database credential. If `None`, the default credential will be used.

polling_interval : int or float, optional

Number of seconds to wait between checks for job completion.

archive : bool, optional (deprecated)

If `True`, archive the import job as soon as it completes.

hidden : bool, optional

If `True` (the default), this job will not appear in the Civis UI.

****kwargs** : kwargs

Extra keyword arguments will be passed to `pandas.DataFrame.to_csv()`.

Returns `fut` : *CivisFuture*

A *CivisFuture* object.

Examples

```
>>> import pandas as pd
>>> df = pd.DataFrame({'a': [1, 2, 3], 'b': [4, 5, 6]})
>>> fut = civis.io.dataframe_to_civis(df, 'my-database',
...                                  'scratch.df_table')
>>> fut.result()
```

civis.io.read_civis

`civis.io.read_civis(table, database, columns=None, use_pandas=False, job_name=None, api_key=None, client=None, credential_id=None, polling_interval=None, archive=False, hidden=True, **kwargs)`

Read data from a Civis table.

Parameters `table` : str

Name of table, including schema, in the database. I.e. `'my_schema.my_table'`.

database : str or int

Read data from this database. Can be the database name or ID.

columns : list, optional

A list of column names. Column SQL transformations are possible. If omitted, all columns are exported.

use_pandas : bool, optional

If `True`, return a `pandas.DataFrame`. Otherwise, return a list of results from `csv.reader()`.

job_name : str, optional

A name to give the job. If omitted, a random job name will be used.

api_key : DEPRECATED str, optional

Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

client : *civis.APIClient*, optional

If not provided, an `civis.APIClient` object will be created from the `CIVIS_API_KEY`.

credential_id : str or int, optional

The database credential ID. If `None`, the default credential will be used.

polling_interval : int or float, optional

Number of seconds to wait between checks for query completion.

archive : bool, optional (deprecated)

If `True`, archive the import job as soon as it completes.

hidden : bool, optional

If `True` (the default), this job will not appear in the Civis UI.

****kwargs** : kwargs

Extra keyword arguments are passed into `pandas.read_csv()` if `use_pandas` is `True` or passed into `csv.reader()` if `use_pandas` is `False`.

Returns **data** : `pandas.DataFrame` or list

A list of rows (with header as first row) if `use_pandas` is `False`, otherwise a `pandas.DataFrame`. Note that if `use_pandas` is `False`, no parsing of types is performed and each row will be a list of strings.

Raises **ImportError**

If `use_pandas` is `True` and `pandas` is not installed.

See also:

`civis.io.read_civis_sql` Read directly into memory using SQL.

`civis.io.civis_to_csv` Write directly to csv.

Examples

```
>>> table = "schema.table"
>>> database = "my_data"
>>> columns = ["column_a", "ROW_NUMBER() OVER(ORDER BY date) AS order"]
>>> data = read_civis(table, database, columns=columns)
>>> columns = data.pop(0)
>>> col_a_index = columns.index("column_a")
>>> col_a = [row[col_a_index] for row in data]
```

```
>>> df = read_civis("schema.table", "my_data", use_pandas=True)
>>> col_a = df["column_a"]
```

`civis.io.read_civis_sql`

`civis.io.read_civis_sql(sql, database, use_pandas=False, job_name=None, api_key=None, client=None, credential_id=None, polling_interval=None, archive=False, hidden=True, **kwargs)`

Read data from Civis using a custom SQL string.

The custom SQL string will be executed twice; once to attempt to retrieve headers and once to retrieve the data. This is done to use a more performant method for retrieving the data. The first execution of the custom SQL is controlled such that changes in state cannot occur (e.g., INSERT, UPDATE, DELETE, etc.).

Parameters `sql` : str, optional

The SQL select string to be executed.

database : str or int

Execute the query against this database. Can be the database name or ID.

use_pandas : bool, optional

If `True`, return a `pandas.DataFrame`. Otherwise, return a list of results from `csv.reader()`.

job_name : str, optional

A name to give the job. If omitted, a random job name will be used.

api_key : DEPRECATED str, optional

Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

client : `civis.APIClient`, optional

If not provided, an `civis.APIClient` object will be created from the `CIVIS_API_KEY`.

credential_id : str or int, optional

The database credential ID. If `None`, the default credential will be used.

polling_interval : int or float, optional

Number of seconds to wait between checks for query completion.

archive : bool, optional (deprecated)

If `True`, archive the import job as soon as it completes.

hidden : bool, optional

If `True` (the default), this job will not appear in the Civis UI.

****kwargs** : kwargs

Extra keyword arguments are passed into `pandas.read_csv()` if `use_pandas` is `True` or passed into `csv.reader()` if `use_pandas` is `False`.

Returns `data` : `pandas.DataFrame` or list

A list of rows (with header as first row) if `use_pandas` is `False`, otherwise a `pandas.DataFrame`. Note that if `use_pandas` is `False`, no parsing of types is performed and each row will be a list of strings.

Raises `ImportError`

If `use_pandas` is `True` and `pandas` is not installed.

See also:

`civis.io.read_civis` Read directly into memory without SQL.

`civis.io.civis_to_csv` Write directly to a CSV file.

Notes

This reads the data into memory.

Examples

```
>>> sql = "SELECT * FROM schema.table"
>>> df = read_civis_sql(sql, "my_database", use_pandas=True)
>>> col_a = df["column_a"]
```

```
>>> data = read_civis_sql(sql, "my_database")
>>> columns = data.pop(0)
>>> col_a_index = columns.index("column_a")
>>> col_a = [row[col_a_index] for row in data]
```

5.2.2 Files

These functions will pass flat files to and from Civis. This is useful if you have data stored in binary or JSON format. Any type of file can be stored in platform via the files endpoint.

<code>civis_to_file(file_id, buf[, api_key, client])</code>	Download a file from Civis.
<code>file_id_from_run_output(name, job_id, run_id)</code>	Find the file ID of a File run output with the name “name”
<code>file_to_civis(buf, name[, api_key, client])</code>	Upload a file to Civis.
<code>file_to_dataframe(file_id[, compression, client])</code>	Load a <code>DataFrame</code> from a CSV stored in a Civis File
<code>file_to_json(file_id[, client])</code>	Restore JSON stored in a Civis File

civis.io.civis_to_file

`civis.io.civis_to_file(file_id, buf, api_key=None, client=None)`

Download a file from Civis.

Parameters `file_id` : int

The Civis file ID.

buf : file-like object or str

A buffer or path specifying where to write the contents of the Civis file. Strings will be treated as paths to local files to open.

api_key : DEPRECATED str, optional

Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

client : `civis.APIClient`, optional

If not provided, an `civis.APIClient` object will be created from the `CIVIS_API_KEY`.

Returns None

Examples

```
>>> file_id = 100
>>> # Download a file to a path on the local filesystem.
>>> civis_to_file(file_id, "my_file.txt")
>>> # Download a file to a file object.
>>> with open("my_file.txt", "wb") as f:
...     civis_to_file(file_id, f)
>>> # Download a file as a bytes object.
>>> import io
>>> buf = io.BytesIO()
>>> civis_to_file(file_id, buf)
>>> # Note that s could be converted to a string with s.decode('utf-8').
>>> s = buf.read()
```

civis.io.file_id_from_run_output

`civis.io.file_id_from_run_output` (*name*, *job_id*, *run_id*, *regex=False*, *client=None*)

Find the file ID of a File run output with the name “name”

The run output is required to have type “File”. If using an approximate match and multiple names match the provided string, return only the first file ID.

Parameters *name* : str

The “name” field of the run output you wish to retrieve

job_id : int

run_id : int

regex : bool, optional

If False (the default), require an exact string match between *name* and the name of the run output. If True, search for a name which matches the regular expression *name* and retrieve the first found.

client : *civis.APIClient*, optional

If not provided, an *civis.APIClient* object will be created from the `CIVIS_API_KEY`.

Returns *file_id* : int

The ID of a Civis File with name matching *name*

Raises *IOError*

If the provided job ID and run ID combination can’t be found

FileNotFoundError

If the run exists, but *name* isn’t in its run outputs

See also:

`APIClient.scripts.list_containers.runs_outputs`

civis.io.file_to_civis

`civis.io.file_to_civis` (*buf*, *name*, *api_key=None*, *client=None*, ***kwargs*)

Upload a file to Civis.

Parameters *buf* : file-like object or str

The file or other buffer that you wish to upload. Strings will be treated as paths to local files to open.

name : str

The name you wish to give the file.

api_key : DEPRECATED str, optional

Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

client : `civis.APIClient`, optional

If not provided, an `civis.APIClient` object will be created from the `CIVIS_API_KEY`.

****kwargs** : kwargs

Extra keyword arguments will be passed to the file creation endpoint. See `post()`.

Returns *file_id* : int

The new Civis file ID.

Notes

If you are opening a binary file (e.g., a compressed archive) to pass to this function, do so using the `'rb'` (read binary) mode (e.g., `open('myfile.zip', 'rb')`).

Warning: If the file-like object is seekable, the current position will be reset to 0.

This facilitates retries and is used to chunk files for multipart uploads for improved performance.

Small or non-seekable file-like objects will be uploaded with a single post.

Examples

```

>>> # Upload file at a given path on the local filesystem.
>>> file_id = file_to_civis("my_data.csv", 'my_data')
>>> # Upload file which expires in 30 days
>>> with open("my_data.csv", "r") as f:
...     file_id = file_to_civis(f, 'my_data')
>>> # Upload file which never expires
>>> with open("my_data.csv", "r") as f:
...     file_id = file_to_civis(f, 'my_data', expires_at=None)

```

civis.io.file_to_dataframe

`civis.io.file_to_dataframe` (*file_id*, *compression='infer'*, *client=None*, ***read_kwargs*)

Load a `DataFrame` from a CSV stored in a Civis File

The `DataFrame` will be read directly from Civis without copying the CSV to a local file on disk.

Parameters `file_id` : int

ID of a Civis File which contains a CSV

compression : str, optional

If “infer”, set the `compression` argument of `pandas.read_csv` based on the file extension of the name of the Civis File. Otherwise pass this argument to `pandas.read_csv`.

client : `civis.APIClient`, optional

If not provided, an `civis.APIClient` object will be created from the `CIVIS_API_KEY`.

****read_kwargs**

Additional arguments will be passed directly to `read_csv()`.

Returns `DataFrame` containing the contents of the CSV

Raises `ImportError`

If `pandas` is not available

See also:

`pandas.read_csv`

`civis.io.file_to_json`

`civis.io.file_to_json` (*file_id*, *client=None*, ***json_kwargs*)

Restore JSON stored in a Civis File

Parameters `file_id` : int

ID of a JSON-formatted Civis File

client : `civis.APIClient`, optional

If not provided, an `civis.APIClient` object will be created from the `CIVIS_API_KEY`.

****json_kwargs**

Additional keyword arguments will be passed directly to `json.load()`.

Returns The object extracted from the JSON-formatted file

See also:

`civis_to_file()`, `json.load()`

5.2.3 Databases

These functions move data from one database to another and expose an interface to run SQL in the database. Use `query_civis()` when you need to execute SQL that does not return data (for example, a `GRANT` or `DROP TABLE` statement).

<code>transfer_table</code> (<i>source_db</i> , <i>dest_db</i> , ...[, ...])

Transfer a table from one location to another.
--

Continued on next page

Table 5.3 – continued from previous page

<code>query_civis(sql, database[, api_key, ...])</code>	Execute a SQL statement as a Civis query.
---	---

civis.io.transfer_table

```
civis.io.transfer_table(source_db, dest_db, source_table, dest_table, job_name=None,
                        api_key=None, client=None, source_credential_id=None,
                        dest_credential_id=None, polling_interval=None, **advanced_options)
```

Transfer a table from one location to another.

Parameters **source_db** : str or int

The name of the database where the source table is located. Optionally, could be the database ID.

dest_db : str or int

The name of the database where the table will be transferred. Optionally, could be the database ID.

source_table : str

Full name of the table to transfer, e.g., 'schema.table'.

dest_table : str

Full name of the table in the destination database, e.g., 'schema.table'.

job_name : str, optional

A name to give the job. If omitted, a random job name will be used.

api_key : DEPRECATED str, optional

Your Civis API key. If not given, the CIVIS_API_KEY environment variable will be used.

client : `civis.APIClient`, optional

If not provided, an `civis.APIClient` object will be created from the CIVIS_API_KEY.

source_credential_id : str or int, optional

Optional credential ID for the source database. If `None`, the default credential will be used.

dest_credential_id : str or int, optional

Optional credential ID for the destination database. If `None`, the default credential will be used.

polling_interval : int or float, optional

Number of seconds to wait between checks for job completion.

****advanced_options** : kwargs

Extra keyword arguments will be passed to the import sync job. See `post_syncs()`.

Returns **results** : `CivisFuture`

A `CivisFuture` object.

Examples

```
>>> transfer_table(source_db='Cluster A', dest_db='Cluster B',
...               source_table='schema.tbl', dest_table='schema.tbl')
```

civis.io.query_civis

`civis.io.query_civis`(*sql*, *database*, *api_key=None*, *client=None*, *credential_id=None*, *preview_rows=10*, *polling_interval=None*, *hidden=True*)

Execute a SQL statement as a Civis query.

Run a query that may return no results or where only a small preview is required. To execute a query that returns a large number of rows, see `read_civis_sql()`.

Parameters *sql* : str

The SQL statement to execute.

database : str or int

The name or ID of the database.

api_key : DEPRECATED str, optional

Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

client : `civis.APIClient`, optional

If not provided, an `civis.APIClient` object will be created from the `CIVIS_API_KEY`.

credential_id : str or int, optional

The ID of the database credential. If `None`, the default credential will be used.

preview_rows : int, optional

The maximum number of rows to return. No more than 100 rows can be returned at once.

polling_interval : int or float, optional

Number of seconds to wait between checks for query completion.

hidden : bool, optional

If `True` (the default), this job will not appear in the Civis UI.

Returns *results* : `CivisFuture`

A `CivisFuture` object.

Examples

```
>>> run = query_civis(sql="DELETE schema.table", database='database')
>>> run.result() # Wait for query to complete
```


5.3 Machine Learning

CivisML uses the Civis Platform to train machine learning models and parallelize their predictions over large datasets. It contains best-practice models for general-purpose classification and regression modeling as well as model quality evaluations and visualizations. All CivisML models use [scikit-learn](#) for interoperability with other platforms and to allow you to leverage resources in the open-source software community when creating machine learning models.

5.3.1 Define Your Model

Start the modeling process by defining your model. Do this by creating an instance of the [ModelPipeline](#) class. Each [ModelPipeline](#) corresponds to a scikit-learn [Pipeline](#) which will run in Civis Platform. A [Pipeline](#) allows you to combine multiple modeling steps (such as missing value imputation and feature selection) into a single model. The [Pipeline](#) is treated as a unit – for example, cross-validation happens over all steps together.

You can define your model in two ways, either by selecting a pre-defined algorithm or by providing your own scikit-learn [Pipeline](#) or [BaseEstimator](#) object. Note that whichever option you chose, CivisML will pre-process your data using either its default ETL, or ETL that you provide (see [Custom ETL](#)).

Pre-Defined Models

You can use the following pre-defined models with CivisML. All models start by imputing missing values with the mean of non-null values in a column. The “sparse_*” models include a LASSO regression step (using the [glmnet](#) package) to do feature selection before passing data to the final model. In some models, CivisML uses default parameters different from those in scikit-learn, as indicated in the “Altered Defaults” column. All models also have `random_state=42`.

Name	Model Type	Algorithm	Altered Defaults
sparse_logistic	classification	LogisticRegression	<code>C=499999950, tol=1e-08</code>
gradient_boosting_classifier	classification	GradientBoostingClassifier	<code>n_estimators=500, max_depth=2</code>
random_forest_classifier	classification	RandomForestClassifier	<code>n_estimators=500</code>
extra_trees_classifier	classification	ExtraTreesClassifier	<code>n_estimators=500</code>
multi-layer_perceptron_classifier	classification	MLPClassifier	
stacking_classifier	classification	StackedClassifier	
sparse_linear_regressor	regression	LinearRegression	
sparse_ridge_regressor	regression	Ridge	
gradient_boosting_regressor	regression	GradientBoostingRegressor	<code>n_estimators=500, max_depth=2</code>
random_forest_regressor	regression	RandomForestRegressor	<code>n_estimators=500</code>
extra_trees_regressor	regression	ExtraTreesRegressor	<code>n_estimators=500</code>
multi-layer_perceptron_regressor	regression	MLPRegressor	
stacking_regressor	regression	StackedRegressor	

The “stacking_classifier” model stacks together the “sparse_logistic”, “gradient_boosting_classifier”, and “ran-

dom_forest_classifier” models, using altered defaults as listed for each in the “Altered Defaults” column of the table above. The models are combined using a `Pipeline` containing a `Normalizer` step, followed by `LogisticRegressionCV` with `penalty='l2'` and `tol=1e-08`. The “stacking_regressor” works similarly, stacking together the “sparse_linear_regressor”, “gradient_boosting_regressor”, and “random_forest_regressor” models, and combining them using `NonNegativeLinearRegression`. The estimators that are being stacked have the same names as the associated pre-defined models, and the meta-estimator steps are named “meta-estimator”. Note that although default parameters are provided for multilayer perceptron models, it is highly recommended that multilayer perceptrons be run using hyperband.

Custom Models

You can create your own `Pipeline` instead of using one of the pre-defined ones. Create the object and pass it as the `model` parameter of the `ModelPipeline`. Your model must follow the scikit-learn API, and you will need to include any dependencies as *Custom Dependencies* if they are not already installed in CivisML. Preinstalled libraries available for your use include:

- `scikit-learn` v0.19.1
- `glmnet` v2.0.0
- `xgboost` v0.6a2
- `muffnn` v1.2.0
- `civismml-extensions` v0.1.6

When you’re assembling your own model, remember that you’ll have to make certain that either you add a missing value imputation step or that your data doesn’t have any missing values. If you’re making a classification model, the model must have a `predict_proba` method. If the class you’re using doesn’t have a `predict_proba` method, you can add one by wrapping it in a `CalibratedClassifierCV`.

Custom ETL

By default, CivisML pre-processes data using the `DataFrameETL` class, with `cols_to_drop` equal to the `excluded_columns` parameter. You can replace this with your own ETL by creating an object of class `BaseEstimator` and passing it as the `etl` parameter during training.

By default, `DataFrameETL` automatically one-hot encodes all categorical columns in the dataset. If you are passing a custom ETL estimator, you will have to ensure that no categorical columns remain after the `transform` method is called on the dataset.

Hyperparameter Tuning

You can tune hyperparameters using one of two methods: grid search or hyperband. CivisML will perform grid search if you pass a dictionary of hyperparameters to the `cross_validation_parameters` parameter, where the keys are hyperparameter names, and the values are lists of hyperparameter values to grid search over. You can run hyperparameter tuning in parallel by setting the `n_jobs` parameter to however many jobs you would like to run in parallel. By default, `n_jobs` is dynamically calculated based on the resources available on your cluster, such that a modeling job will never take up more than 90% of the cluster resources at once.

Hyperband is an efficient approach to hyperparameter optimization, and *recommended over grid search where possible*. CivisML will perform hyperband optimization for a pre-defined model if you pass the string 'hyperband' to `cross_validation_parameters`. Hyperband is currently only supported for the following models: `gradient_boosting_classifier`, `random_forest_classifier`, `extra_trees_classifier`, `multilayer_perceptron_classifier`, `stacking_classifier`,

`gradient_boosting_regressor`, `random_forest_regressor`, `extra_trees_regressor`, `multilayer_perceptron_regressor`, and `stacking_regressor`.

Hyperband cannot be used to tune GLMs. For this reason, preset GLMs do not have a hyperband option. Similarly, when `cross_validation_parameters='hyperband'` and the model is `stacking_classifier` or `stacking_regressor`, only the GBT and random forest steps of the stacker are tuned using hyperband. Note that if you want to use hyperband with a custom model, you will need to wrap your estimator in a `civismlex`. `hyperband.HyperbandSearchCV` estimator yourself.

CivisML runs pre-defined models with hyperband using the following distributions:

Models	Cost Parameter	Hyperband Distributions
<code>gradient_boosting_classifier</code> <code>gradient_boosting_regressor</code> GBT step in <code>stacking_classifier</code> GBT step in <code>stacking_regressor</code>	<code>n_estimators</code> <code>min = 100,</code> <code>max = 1000</code>	<code>max_depth:</code> <code>randint(low=1, high=5)</code> <code>max_features:</code> <code>[None,</code> <code>'sqrt', 'log2', 0.5,</code> <code>0.3, 0.1, 0.05, 0.01]</code> <code>learning_rate:</code> <code>truncexpon(b=5,</code> <code>loc=.0003,</code> <code>scale=1./167.)</code>
<code>random_forest_classifier</code> <code>random_forest_regressor</code> <code>extra_trees_classifier</code> <code>extra_trees_regressor</code> RF step in <code>stacking_classifier</code> RF step in <code>stacking_regressor</code>	<code>n_estimators</code> <code>min = 100,</code> <code>max = 1000</code>	<code>criterion:</code> <code>['gini',</code> <code>'entropy']</code> <code>max_features:</code> <code>truncexpon(b=10.,</code> <code>loc=.01, scale=1./10.11)</code> <code>max_depth:</code> <code>[1, 2, 3, 4,</code> <code>6, 10, None]</code>
<code>multilayer_perceptron_classifier</code> <code>multilayer_perceptron_regressor</code>	<code>n_epochs</code> <code>min = 5,</code> <code>max = 50</code>	<code>keep_prob:</code> <code>uniform()</code> <code>hidden_units:</code> <code>[(),</code> <code>(16,), (32,), (64,),</code> <code>(64, 64), (64, 64, 64),</code> <code>(128,), (128, 128),</code> <code>(128, 128, 128),</code> <code>(256,),</code> <code>(256, 256), (256,</code> <code>256, 256), (512,</code> <code>256, 128, 64),</code> <code>(1024, 512, 256,</code> <code>128)]</code> <code>learning_rate:</code> <code>[1e-2,</code> <code>2e-2, 5e-2, 8e-2, 1e-3,</code> <code>2e-3, 5e-3, 8e-3, 1e-4]</code>

The truncated exponential distribution for the gradient boosting classifier and regressor was chosen to skew the dis-

tribution toward small values, ranging between .0003 and .03, with a mean close to .006. Similarly, the truncated exponential distribution for the random forest and extra trees models skews toward small values, ranging between .01 and 1, and with a mean close to .1.

Custom Dependencies

Installing packages from PyPI is straightforward. You can specify a *dependencies*

argument to *ModelPipeline* which will install the dependencies in your runtime environment. VCS support is also enabled (see [docs](#).) Installing a remote git repository from, say, Github only requires passing the HTTPS URL in the form of, for example, `git+https://github.com/scikit-learn/scikit-learn`.

CivisML will run `pip install [your package here]`. We strongly encourage you to pin package versions for consistency. Example code looks like:

```
from civis.ml import ModelPipeline
from pyearth import Earth
deps = ['git+https://github.com/scikit-learn-contrib/py-earth.
↳git@da856e11b2a5d16aba07f51c3c15cef5e40550c7']
est = Earth()
model = ModelPipeline(est, dependent_variable='age', dependencies=deps)
train = model.train(table_name='donors.from_march', database_name='client')
```

Additionally, you can store a remote git host's API token in the Civis Platform as a credential to use for installing private git repositories. For example, you can go to Github at the <https://github.com/settings/tokens> URL, copy your token into the password field of a credential, and pass the credential name to the `git_token_name` argument in *ModelPipeline*. This also works with other hosting services. A simple example of how to do this with API looks as follows

```
import civis
password = 'abc123' # token copied from https://github.com/settings/tokens
username = 'user123' # Github username
git_token_name = 'Github credential'

client = civis.APIClient()
credential = client.credentials.post(password=password,
                                     username=username,
                                     name=git_token_name,
                                     type="Custom")

pipeline = civis.ml.ModelPipeline(..., git_token_name=git_token_name)
```

Note, installing private dependencies with submodules is not supported.

5.3.2 Asynchronous Execution

All calls to a *ModelPipeline* object are non-blocking, i.e. they immediately provide a result without waiting for the job in the Civis Platform to complete. Calls to `civis.ml.ModelPipeline.train()` and `civis.ml.ModelPipeline.predict()` return a *ModelFuture* object, which is a subclass of *Future* from the Python standard library. This behavior lets you train multiple models at once, or generate predictions from models, while still doing other work while waiting for your jobs to complete.

The *ModelFuture* can find and retrieve outputs from your CivisML jobs, such as trained *Pipeline* objects or out-of-sample predictions. The *ModelFuture* only downloads outputs when you request them.

5.3.3 Model Persistence

Civis Platform permanently stores all models, indexed by the job ID and the run ID (also called a “build”) of the training job. If you wish to use an existing model, call `civis.ml.ModelPipeline.from_existing()` with the job ID of the training job. You can find the job ID with the `train_job_id` attribute of a `ModelFuture`, or by looking at the URL of your model on the [Civis Platform models page](#). If the training job has multiple runs, you may also provide a run ID to select a run other than the most recent. You can list all model runs of a training job by calling `civis.APIClient().jobs.get(train_job_id)['runs']`. You may also store the `ModelPipeline` itself with the `pickle` module.

5.3.4 Examples

`Future` objects have the method `add_done_callback()`. This is called as soon as the run completes. It takes a single argument, the `Future` for the completed job. You can use this method to chain jobs together:

```
from concurrent import futures
from civis.ml import ModelPipeline
import pandas as pd
df = pd.read_csv('data.csv')
training, predictions = [], []
model = ModelPipeline('sparse_logistic', dependent_variable='type')
training.append(model.train(df))
training[-1].add_done_callback(lambda fut: predictions.append(model.predict(df)))
futures.wait(training) # Blocks until all training jobs complete
futures.wait(predictions) # Blocks until all prediction jobs complete
```

You can create and train multiple models at once to find the best approach for solving a problem. For example:

```
from civis.ml import ModelPipeline
algorithms = ['gradient_boosting_classifier', 'sparse_logistic', 'random_forest_
↳ classifier']
pkey = 'person_id'
depvar = 'likes_cats'
models = [ModelPipeline(alg, primary_key=pkey, dependent_variable=depvar) for alg in
↳ algorithms]
train = [model.train(table_name='schema.name', database_name='My DB') for model in
↳ models]
aucs = [tr.metrics['roc_auc'] for tr in train] # Code blocks here
```

5.3.5 Optional dependencies

You do not need any external libraries installed to use CivisML, but the following pip-installable dependencies enhance the capabilities of the `ModelPipeline`:

- pandas
- scikit-learn
- glmnet
- feather-format
- civismml-extensions
- mufnn

Install `pandas` if you wish to download tables of predictions. You can also model on `DataFrame` objects in your interpreter.

If you wish to use the `ModelPipeline` code to model on `DataFrame` objects in your local environment, the `feather-format` package (requires `pandas >= 0.20`) will improve data transfer speeds and guarantee that your data types are correctly detected by CivisML. You must install `feather-format` if you wish to use `pd.Categorical` columns in your `DataFrame` objects, since that type information is lost when writing data as a CSV.

If you wish to use custom models or download trained models, you'll need scikit-learn installed.

Several pre-defined models rely on public Civis Analytics libraries. The “`sparse_logistic`”, “`sparse_linear_regressor`”, “`sparse_ridge_regressor`”, “`stacking_classifier`”, and “`stacking_regressor`” models all use the `glmnet` library. Pre-defined MLP models (“`multilayer_perceptron_classifier`” and “`multilayer_perceptron_regressor`”) depend on the `muffnn` library. Finally, models which use the default CivisML ETL, along with models which use stacking or hyperband, depend on `civismml-extensions`. Install these packages if you wish to download the pre-defined models that depend on them.

5.3.6 Object reference

```
class civis.ml.ModelPipeline(model,      dependent_variable,      primary_key=None,      pa-
                             rameters=None,      cross_validation_parameters=None,
                             model_name=None, calibration=None, excluded_columns=None,
                             client=None, cpu_requested=None, memory_requested=None,
                             disk_requested=None, notifications=None, dependencies=None,
                             git_token_name=None, verbose=False, etl=None)
```

Interface for scikit-learn modeling in the Civis Platform

Each `ModelPipeline` corresponds to a scikit-learn `Pipeline` which will run in Civis Platform.

Note that this object can be safely pickled and unpickled, but it does not store the state of any attached `APIClient` object. An unpickled `ModelPipeline` will use the API key from the user's environment.

Parameters `model` : string or Estimator

Either the name of a pre-defined model (e.g. “`sparse_logistic`” or “`gradient_boosting_classifier`”) or else a pre-existing Estimator object.

dependent_variable : string or List[str]

The dependent variable of the training dataset. For a multi-target problem, this should be a list of column names of dependent variables. Nulls in a single dependent variable will automatically be dropped.

primary_key : string, optional

The unique ID (primary key) of the training dataset. This will be used to index the out-of-sample scores.

parameters : dict, optional

Specify parameters for the final stage estimator in a predefined model, e.g. `{ 'C' : 2 }` for a “`sparse_logistic`” model.

cross_validation_parameters : dict or string, optional

Options for cross validation. For grid search, supply a parameter grid as a dictionary, e.g., `{{ 'n_estimators': [100, 200, 500], 'learning_rate': [0.01, 0.1], 'max_depth': [2, 3] }}`. For hyperband, pass the string “`hyperband`”.

model_name : string, optional

The prefix of the Platform modeling jobs. It will have "Train" or "Predict" added to become the Script title.

calibration : {None, "sigmoid", "isotonic"}

If not None, calibrate output probabilities with the selected method. Valid only with classification models.

excluded_columns : array, optional

A list of columns which will be considered ineligible to be independent variables.

client : *APIClient*, optional

If not provided, an *APIClient* object will be created from the `CIVIS_API_KEY`.

cpu_requested : int, optional

Number of CPU shares requested in the Civis Platform for training jobs. 1024 shares = 1 CPU.

memory_requested : int, optional

Memory requested from Civis Platform for training jobs, in MiB

disk_requested : float, optional

Disk space requested on Civis Platform for training jobs, in GB

notifications : dict

See *post_custom()* for further documentation about email and URL notification.

dependencies : array, optional

List of packages to install from PyPI or git repository (i.e., Github or Bitbucket). If a private repo is specified, please include a `git_token_name` argument as well (see below). Make sure to pin dependencies to a specific version, since dependencies will be reinstalled during every training and predict job.

git_token_name : str, optional

Name of remote git API token stored in Civis Platform as the password field in a custom platform credential. Used only when installing private git repositories.

verbose : bool, optional

If True, supply debug outputs in Platform logs and make prediction child jobs visible.

etl : Estimator, optional

Custom ETL estimator which overrides the default ETL, and is run before training and validation.

See also:

civis.ml.ModelFuture

Examples

```
>>> from civis.ml import ModelPipeline
>>> model = ModelPipeline('gradient_boosting_classifier', 'depvar',
...                       primary_key='voterbase_id')
>>> train = model.train(table_name='schema.survey_data',
...                     fit_params={'sample_weight': 'survey_weight'},
```

```

...             database_name='My Redshift Cluster',
...             oos_scores='scratch.survey_depvar_oos_scores')
>>> train
<ModelFuture at 0x11be7ae10 state=queued>
>>> train.running()
True
>>> train.done()
False
>>> df = train.table # Read OOS scores from its Civis File. Blocking.
>>> meta = train.metadata # Metadata from training run
>>> train.metrics['roc_auc']
0.88425
>>> pred = model.predict(table_name='schema.demographics_table ',
...                       database_name='My Redshift Cluster',
...                       output_table='schema.predicted_survey_response',
...                       if_exists='drop')
>>> df_pred = pred.table # Blocks until finished
# Modify the parameters of the base estimator in a default model:
>>> model = ModelPipeline('sparse_logistic', 'depvar',
...                       primary_key='voterbase_id',
...                       parameters={'C': 2})
# Grid search over hyperparameters in the base estimator:
>>> model = ModelPipeline('sparse_logistic', 'depvar',
...                       primary_key='voterbase_id',
...                       cross_validation_parameters={'C': [0.1, 1, 10]})

```

Attributes

estimator	(Pipeline) The trained scikit-learn Pipeline
train_result_	(ModelFuture) ModelFuture encapsulating this model's training run
state	(str) Status of the training job (non-blocking)

Methods

train()	Train the model on data in Civis Platform; outputs <i>ModelFuture</i>
predict()	Make predictions on new data; outputs <i>ModelFuture</i>
from_existing()	Class method; use to create a <i>ModelPipeline</i> from an existing model training run

classmethod from_existing (*train_job_id*, *train_run_id*='latest', *client*=None)

Create a *ModelPipeline* object from existing model IDs

Parameters *train_job_id* : int

The ID of the CivisML job in the Civis Platform

train_run_id : int or string, optional

Location of the model run, either

- an explicit run ID,
- “latest” : The most recent run
- “active” : The run designated by the training job’s “active build” parameter

client : *APIClient*, optional

If not provided, an *APIClient* object will be created from the `CIVIS_API_KEY`.

Returns *ModelPipeline*

A *ModelPipeline* which refers to a previously-trained model

Examples

```
>>> from civis.ml import ModelPipeline
>>> model = ModelPipeline.from_existing(job_id)
>>> model.train_result_.metrics['roc_auc']
0.843
```

predict (*df=None*, *csv_path=None*, *table_name=None*, *database_name=None*, *manifest=None*, *file_id=None*, *sql_where=None*, *sql_limit=None*, *primary_key=Sentinel()*, *output_table=None*, *output_db=None*, *if_exists='fail'*, *n_jobs=None*, *polling_interval=None*, *cpu=None*, *memory=None*, *disk_space=None*)

Make predictions on a trained model

Provide input through one of a *DataFrame* (*df*), a local CSV (*csv_path*), a Civis Table (*table_name* and *database_name*), a Civis File containing a CSV (*file_id*), or a Civis File containing a manifest file (*manifest*).

A “manifest file” is JSON which specifies the location of many shards of the data to be used for prediction. A manifest file is the output of a Civis export job with `force_multifile=True` set, e.g. from `civis.io.civis_to_multifile_csv()`. Large Civis Tables (provided using *table_name*) will automatically be exported to manifest files.

Prediction outputs will always be stored as gzipped CSVs in one or more Civis Files. You can find a list of File ID numbers for output files at the “output_file_ids” key in the metadata returned by the prediction job. Provide an *output_table* (and optionally an *output_db*, if it’s different from *database_name*) to copy these predictions into a Civis Table.

Parameters *df* : *pd.DataFrame*, optional

A *DataFrame* of data for prediction. The *DataFrame* will be uploaded to a Civis file so that CivisML can access it. Note that the index of the *DataFrame* will be ignored – use `df.reset_index()` if you want your index column to be included with the data passed to CivisML. NB: You must install `feather-format` if your *DataFrame* contains *Categorical* columns, to ensure that CivisML preserves data types.

csv_path : str, optional

The location of a CSV of data on the local disk. It will be uploaded to a Civis file.

table_name : str, optional

The qualified name of the table containing your data

database_name : str, optional

Name of the database holding the data, e.g., ‘My Redshift Cluster’.

manifest : int, optional

ID for a manifest file stored as a Civis file. (Note: if the manifest is not a Civis Platform-specific manifest, like the one returned from `civis.io.civis_to_multifile_csv()`, this must be used in conjunction with *table_name* and *database_name* due to the need for column discovery via Redshift.)

file_id : int, optional

If the data are a CSV stored in a Civis file, provide the integer file ID.

sql_where : str, optional

A SQL WHERE clause used to scope the rows to be predicted

sql_limit : int, optional

SQL LIMIT clause to restrict the size of the prediction set

primary_key : str, optional

Primary key of the prediction table. Defaults to the primary key of the training data. Use `None` to indicate that the prediction data don't have a primary key column.

output_table: str, optional

The table in which to put the predictions.

output_db : str, optional

Database of the output table. Defaults to the database of the input table.

if_exists : {'fail', 'append', 'drop', 'truncate'}

Action to take if the prediction table already exists.

n_jobs : int, optional

Number of concurrent Platform jobs to use for multi-file / large table prediction. Defaults to *None*, which allows CivisML to dynamically calculate an appropriate number of workers to use (in general, as many as possible without using all resources in the cluster).

polling_interval : float, optional

Check for job completion every this number of seconds. Do not set if using the notifications endpoint.

cpu : int, optional

CPU shares requested by the user for a single job.

memory : int, optional

RAM requested by the user for a single job.

disk_space : float, optional

disk space requested by the user for a single job.

Returns *ModelFuture*

train (*df=None*, *csv_path=None*, *table_name=None*, *database_name=None*, *file_id=None*, *sql_where=None*, *sql_limit=None*, *oos_scores=None*, *oos_scores_db=None*, *if_exists='fail'*, *fit_params=None*, *polling_interval=None*, *validation_data='train'*, *n_jobs=None*)
Start a Civis Platform job to train your model

Provide input through one of a *DataFrame* (*df*), a local CSV (*csv_path*), a Civis Table (*table_name* and *database_name*), or a Civis File containing a CSV (*file_id*).

Model outputs will always contain out-of-sample scores (accessible through *ModelFuture.table* on this function's output), and you may chose to store these out-of-sample scores in a Civis Table with the *oos_scores*, *oos_scores_db*, and *if_exists* parameters.

Parameters *df* : *pd.DataFrame*, optional

A `DataFrame` of training data. The `DataFrame` will be uploaded to a Civis file so that CivisML can access it. Note that the index of the `DataFrame` will be ignored – use `df.reset_index()` if you want your index column to be included with the data passed to CivisML. NB: You must install `feather-format` if your `DataFrame` contains `Categorical` columns, to ensure that CivisML preserves data types.

csv_path : str, optional

The location of a CSV of data on the local disk. It will be uploaded to a Civis file.

table_name : str, optional

The qualified name of the table containing the training set from which to build the model.

database_name : str, optional

Name of the database holding the training set table used to build the model. E.g., ‘My Cluster Name’.

file_id : int, optional

If the training data are stored in a Civis file, provide the integer file ID.

sql_where : str, optional

A SQL WHERE clause used to scope the rows of the training set (used for table input only)

sql_limit : int, optional

SQL LIMIT clause for querying the training set (used for table input only)

oos_scores : str, optional

If provided, store out-of-sample predictions on training set data to this Redshift “schema.tablename”.

oos_scores_db : str, optional

If not provided, store OOS predictions in the same database which holds the training data.

if_exists : {‘fail’, ‘append’, ‘drop’, ‘truncate’}

Action to take if the out-of-sample prediction table already exists.

fit_params: Dict[str, str]

Mapping from parameter names in the model’s `fit` method to the column names which hold the data, e.g. { ‘sample_weight’: ‘survey_weight_column’ }.

polling_interval : float, optional

Check for job completion every this number of seconds. Do not set if using the notifications endpoint.

validation_data : str, optional

Source for validation data. There are currently two options: ‘train’ (the default), which cross-validates over training data for validation; and ‘skip’, which skips the validation step.

n_jobs : int, optional

Number of jobs to use for training and validation. Defaults to *None*, which allows CivisML to dynamically calculate an appropriate number of workers to use (in general, as many as possible without using all resources in the cluster). Increase *n_jobs* to parallelize over many hyperparameter combinations in grid search/hyperband, or decrease to use fewer computational resources at once.

Returns *ModelFuture*

```
class civis.ml.ModelFuture(job_id, run_id, train_job_id=None, train_run_id=None,
                             polling_interval=None, client=None, poll_on_creation=True)
```

Encapsulates asynchronous execution of a CivisML job

This object knows where to find modeling outputs from CivisML jobs. All data attributes are lazily retrieved and block on job completion.

This object can be pickled, but it does not store the state of the attached *APIClient* object. An unpickled *ModelFuture* will use the API key from the user's environment.

Parameters *job_id* : int

ID of the modeling job

run_id : int

ID of the modeling run

train_job_id : int, optional

If not provided, this object is assumed to encapsulate a training job, and *train_job_id* will equal *job_id*.

train_run_id : int, optional

If not provided, this object is assumed to encapsulate a training run, and *train_run_id* will equal *run_id*.

polling_interval : int or float, optional

The number of seconds between API requests to check whether a result is ready. The default intelligently switches between a short interval if pubnub is not available and a long interval for pubnub backup if that library is installed.

client : *civis.APIClient*, optional

If not provided, an *civis.APIClient* object will be created from the *CIVIS_API_KEY*.

poll_on_creation : bool, optional

If *True* (the default), it will poll upon calling *result()* the first time. If *False*, it will wait the number of seconds specified in *polling_interval* from object creation before polling.

See also:

civis.futures.CivisFuture, *civis.futures.ContainerFuture*, *concurrent.futures.Future*

Attributes

meta-data	(dict, blocking) The metadata associated with this modeling job
metrics	(dict, blocking) Validation metrics from this job's training
validation_metadata	(dict, blocking) Metadata from this modeling job's validation run
train_metadata	(dict, blocking) Metadata from this modeling job's training run (will be identical to <i>metadata</i> if this is a training run)
estimator	(<code>sklearn.pipeline.Pipeline</code> , blocking) The fitted scikit-learn Pipeline resulting from this model run
table	(<code>pandas.DataFrame</code> , blocking) The table output from this modeling job: out-of-sample predictions on the training set for a training job, or a table of predictions for a prediction job. If the prediction job was split into multiple files (this happens automatically for large tables), this attribute will provide only predictions for the first file.
state	(str) The current state of the Civis Platform run
job_id	(int)
run_id	(int)
train_job_id	(int) Container ID for the training job – identical to <i>job_id</i> if this is a training job.
train_run_id	(int) As <i>train_job_id</i> but for runs
is_training	(bool) True if this <code>ModelFuture</code> corresponds to a train-validate job.

Methods

<code>cancel()</code>	Cancels the corresponding Platform job before completion
<code>succeeded()</code>	(Non-blocking) Is the job a success?
<code>failed()</code>	(Non-blocking) Did the job fail?
<code>cancelled()</code>	(Non-blocking) Was the job cancelled?
<code>running()</code>	(Non-blocking) Is the job still running?
<code>done()</code>	(Non-blocking) Is the job finished?
<code>result()</code>	(Blocking) Return the final status of the Civis Platform job.

`add_done_callback (fn)`

Attaches a callable that will be called when the future finishes.

Args:

fn: A callable that will be called with this future as its only argument when the future completes or is cancelled. The callable will always be called by a thread in the same process in which it was added. If the future has already completed or been cancelled then the callable will be called immediately. These callables are called in the order that they were added.

`cancel ()`

Submit a request to cancel the container/script/run.

Returns bool

Whether or not the job is in a cancelled state.

`cancelled ()`

Return True if the future was cancelled.

done()

Return True if the future was cancelled or finished executing.

exception (*timeout=None*)

Return the exception raised by the call that the future represents.

Args:

timeout: The number of seconds to wait for the exception if the future isn't done. If None, then there is no limit on the wait time.

Returns: The exception raised by the call that the future represents or None if the call completed without raising.

Raises: CanceledError: If the future was cancelled. TimeoutError: If the future didn't finish executing before the given timeout.

failed()

Return True if the Civis job failed.

result (*timeout=None*)

Return the result of the call that the future represents.

Args:

timeout: The number of seconds to wait for the result if the future isn't done. If None, then there is no limit on the wait time.

Returns: The result of the call that the future represents.

Raises: CanceledError: If the future was cancelled. TimeoutError: If the future didn't finish executing before the given timeout.

Exception: If the call raised then that exception will be raised.

running()

Return True if the future is currently executing.

set_exception (*exception*)

Sets the result of the future as being the given exception.

Should only be used by Executor implementations and unit tests.

set_result (*result*)

Sets the return value of work associated with the future.

Should only be used by Executor implementations and unit tests.

set_running_or_notify_cancel()

Mark the future as running or process any cancel notifications.

Should only be used by Executor implementations and unit tests.

If the future has been cancelled (cancel() was called and returned True) then any threads waiting on the future completing (though calls to as_completed() or wait()) are notified and False is returned.

If the future was not cancelled then it is put in the running state (future calls to running() will return True) and True is returned.

This method should be called by Executor implementations before executing the work associated with this future. If this method returns False then the work should not be executed.

Returns: False if the Future was cancelled, True otherwise.

Raises:

RuntimeError: if this method was already called or if `set_result()` or `set_exception()` was called.

succeeded()

Return True if the job completed in Civis with no error.

5.4 Parallel Computation

The Civis Platform manages a pool of cloud computing resources. You can access these resources with the tools in the `civis.parallel` and `civis.futures` modules.

5.4.1 Joblib backend

If you can divide your work into multiple independent chunks, each of which takes at least several minutes to run, you can reduce the time your job takes to finish by running each chunk simultaneously in Civis Platform. The Civis joblib backend is a software tool which makes it easier to run many jobs simultaneously.

Things to keep in mind when deciding if the Civis joblib backend is the right tool for your code:

- Each function call which is parallelized with the Civis joblib backend will run in a different Civis Platform script. Creating a new script comes with some overhead. It will take between a few seconds and a few minutes for each script to start, depending on whether Civis Platform needs to provision additional resources. If you expect that each function call will complete quickly, instead consider either running them in serial or using extra processes in the same Civis Platform script.
- Because function calls run in different scripts, function inputs and outputs must be uploaded to Civis Platform from their origin script and downloaded into their destination. If your functions take very large inputs and/or produce very large outputs, moving the data around will cause additional overhead. Consider either using a different tool or refactoring your code so that the function to be parallelized is no longer moving around large amounts of data.
- Some open-source libraries, such as `scikit-learn`, use `joblib` to do computations in parallel. If you're working with such a library, the Civis joblib backend provides an easy way to run these parallel computations in different Civis Platform scripts.

Joblib

`joblib` is an open source Python library which facilitates parallel processing in Python. `Joblib` uses Python's `multiprocessing` library to run functions in parallel, but it also allows users to define their own “back end” for parallel computation. The Civis Python API client takes advantage of this to let you easily run your own code in parallel through Civis Platform.

The `make_backend_factory()`, `infer_backend_factory()`, and `make_backend_template_factory()` functions allow you to define a “civis” parallel computation backend which will transparently distribute computation in cloud resources managed by the Civis Platform.

See the [joblib user guide](#) for examples of using `joblib` to do parallel computation. Note that the descriptions of “memmapping” aren't relevant to using Civis Platform as a backend, since your jobs will potentially run on different computers and can't share memory. Using the Civis joblib backend to run jobs in parallel in the cloud looks the same as running jobs in parallel on your local computer, except that you first need to set up the “civis” backend.

How to use

Begin by defining the backend. The Civis joblib backend creates and runs Container Scripts, and the `make_backend_factory()` function accepts several arguments which will be passed to `post_containers()`. For example, you could pass a `repo_http_uri` or `repo_ref` to clone a repository from GitHub into the container which will run your function. Use the `docker_image_name` and `docker_image_tag` to select a custom Docker image for your job. You can provide a `setup_cmd` to run setup in bash before your function executes in Python. The default `setup_cmd` will run `python setup.py install` in the base directory of any `repo_http_uri` which you include in your backend setup. Make sure that the environment you define for your Civis backend includes all of the code which your parallel function will call.

The `make_backend_factory()` function will return a backend factory which should be given to the `joblib.register_parallel_backend()` function. For example:

```
>>> from joblib import register_parallel_backend
>>> from civis.parallel import make_backend_factory
>>> be_factory = make_backend_factory()
>>> register_parallel_backend('civis', be_factory)
```

Direct joblib to use a custom backend by entering a `joblib.parallel_backend()` context:

```
>>> from joblib import parallel_backend
>>> with parallel_backend('civis'):
...     # Do joblib parallel computation here.
```

You can find more about custom joblib backends in the [joblib documentation](#).

Note that `joblib.Parallel` takes both a `n_jobs` and `pre_dispatch` parameter. The Civis joblib backend doesn't queue submitted jobs itself, so it will run `pre_dispatch` jobs at once. The default value of `pre_dispatch` is `"2*n_jobs"`, which will run a maximum of $2 * n_jobs$ jobs at once in the Civis Platform. Set `pre_dispatch="n_jobs"` in your `Parallel` call to run at most `n_jobs` jobs.

The Civis joblib backend uses [cloudpickle](#) to transport code and data from the parent environment to the Civis Platform. This means that you may parallelize dynamically-defined functions and classes, including lambda functions.

Infer backend parameters

If you're writing code which will run inside a Civis Container Script, then the `infer_backend_factory()` function returns a backend factory with environment parameters pre-populated by inspecting the state of your container script at run time. Use `infer_backend_factory()` anywhere you would use `make_backend_factory()`, and you don't need to specify a Docker image or GitHub repository.

Templated Scripts

The `make_backend_template_factory()` is intended for developers who are writing code which may be run by users who don't have permissions to create new container scripts with the necessary environment.

Instead of defining and creating new container scripts with `make_backend_factory()`, you can use `make_backend_template_factory()` to launch custom scripts from a templated script. To use the template factory, your backing container script must have the Civis Python client installed, and its run command must finish by calling `civis_joblib_worker` with no arguments. The template must accept the parameter `"JOBLIB_FUNC_FILE_ID"`. The Civis joblib backend will use this parameter to transport your remote work.

Examples

Parallel computation using the default joblib backend (this uses processes on your local computer):

```
>>> def expensive_calculation(num1, num2):
...     return 2 * num1 + num2
>>> from joblib import delayed, Parallel
>>> parallel = Parallel(n_jobs=5)
>>> args = [(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1)]
>>> print(parallel(delayed(expensive_calculation)(*a) for a in args))
[1, 3, 5, 7, 9, 11, 13]
```

You can do the the same parallel computation using the Civis backend by creating and registering a backend factory and entering a with `parallel_backend('civis')` context. The code below will start seven different jobs in Civis Platform (with up to five running at once). Each job will call the function `expensive_calculation` with a different set of arguments from the list `args`:

```
>>> def expensive_calculation(num1, num2):
...     return 2 * num1 + num2
>>> from joblib import delayed, Parallel
>>> from joblib import parallel_backend, register_parallel_backend
>>> from civis.parallel import make_backend_factory
>>> register_parallel_backend('civis', make_backend_factory(
...     required_resources={"cpu": 512, "memory": 256}))
>>> args = [(0, 1), (1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 1)]
>>> with parallel_backend('civis'):
...     parallel = Parallel(n_jobs=5, pre_dispatch='n_jobs')
...     print(parallel(delayed(expensive_calculation)(*a) for a in args))
[1, 3, 5, 7, 9, 11, 13]
```

You can use the Civis joblib backend to parallelize any code which uses joblib internally, such as scikit-learn:

```
>>> from joblib import parallel_backend, register_parallel_backend
>>> from sklearn.model_selection import GridSearchCV
>>> from sklearn.ensemble import GradientBoostingClassifier
>>> from sklearn.datasets import load_digits
>>> digits = load_digits()
>>> param_grid = {
...     "max_depth": [1, 3, 5, None],
...     "max_features": ["sqrt", "log2", None],
...     "learning_rate": [0.1, 0.01, 0.001]
... }
>>> # Note: n_jobs and pre_dispatch specify the maximum number of
>>> # concurrent jobs.
>>> gs = GridSearchCV(GradientBoostingClassifier(n_estimators=1000,
...     random_state=42),
...     param_grid=param_grid,
...     n_jobs=5, pre_dispatch="n_jobs")
>>> register_parallel_backend('civis', make_backend_factory(
...     required_resources={"cpu": 512, "memory": 256}))
>>> with parallel_backend('civis'):
...     gs.fit(digits.data, digits.target)
```

Debugging

Any (non-retried) errors in child jobs will cause the entire parallel call to fail. `joblib` will transport the first exception from a remote job and raise it in the parent process so that you can debug.

If your remote jobs are failing because of network problems (e.g. occasional 500 errors), you can make your parallel call more likely to succeed by using a `max_job_retries` value above 0 when creating your backend factory. This will automatically retry a job (potentially more than once) before giving up and keeping an exception.

Logging: The Civis joblib backend uses the standard library `logging module`, with `debug` emits for events which might help you diagnose errors. See also the “verbose” argument to `joblib.Parallel`, which prints information to either `stdout` or `stderr`.

Mismatches between your local environment and the environment in the Civis container script jobs are a common source of errors. To run a function in the Civis platform, any modules called by that function must be importable from a Python interpreter running in the container script. For example, if you use `joblib.Parallel` with `numpy.sqrt()`, the joblib backend must be set to run your function in a container which has `numpy` installed. If you see an error such as:

```
ModuleNotFoundError: No module named 'numpy'
```

this signifies that the function you’re trying to run doesn’t exist in the remote environment. Select a Docker container with the module installed, or install it in your remote environment by using the `repo_http_uri` parameter of `make_backend_factory()` to install it from GitHub.

5.4.2 Object Reference

Parallel computations using the Civis Platform infrastructure

```
civis.parallel.infer_backend_factory(required_resources=None, params=None, arguments=None, client=None, polling_interval=None, setup_cmd=None, max_submit_retries=0, max_job_retries=0, hidden=True, remote_backend='sequential', **kwargs)
```

Infer the container environment and return a backend factory.

This function helps you run additional jobs from code which executes inside a Civis container job. The function reads settings for relevant parameters (e.g. the Docker image) of the container it’s running inside of.

Note: This function will read the state of the parent container job at the time this function executes. If the user has modified the container job since the run started (e.g. by changing the GitHub branch in the container’s GUI), this function may infer incorrect settings for the child jobs.

Keyword arguments inferred from the existing script’s state are [`‘docker_image_name’`, `‘docker_image_tag’`, `‘repo_http_uri’`, `‘repo_ref’`, `‘remote_host_credential_id’`, `‘git_credential_id’`, `‘cancel_timeout’`, `‘time_zone’`]

Parameters `required_resources` : dict or None, optional

The resources needed by the container. See the *container scripts API documentation* <<https://platform.civisanalytics.com/api#resources-scripts>> for details. Resource requirements not specified will default to the requirements of the current job.

params : list or None, optional

A definition of the parameters this script accepts in the arguments field. See the *container scripts API documentation* <<https://platform.civisanalytics.com/api#resources-scripts>> for details.

Parameters of the child jobs will default to the parameters of the current job. Any parameters provided here will override parameters of the same name from the current job.

arguments : dict or None, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params. See the *container scripts API documentation* <<https://platform.civisanalytics.com/api#resources-scripts>> for details.

Arguments will default to the arguments of the current job. Anything provided here will override portions of the current job's arguments.

client : *civis.APIClient* instance or None, optional

An API Client object to use.

polling_interval : int, optional

The polling interval, in seconds, for checking container script status. If you have many jobs, you may want to set this higher (e.g., 300) to avoid *rate-limiting* <<https://platform.civisanalytics.com/api#basics>>. You should only set this if you aren't using pubnub notifications.

setup_cmd : str, optional

A shell command or sequence of commands for setting up the environment. These will precede the commands used to run functions in joblib. This is primarily for installing dependencies that are not available in the dockerhub repo (e.g., "cd /app && python setup.py install" or "pip install gensim").

With no GitHub repo input, the setup command will default to a command that does nothing. If a `repo_http_uri` is provided, the default setup command will attempt to run "python setup.py install". If this command fails, execution will still continue.

max_submit_retries : int, optional

The maximum number of retries for submitting each job. This is to help avoid a large set of jobs failing because of a single 5xx error. A value higher than zero should only be used for jobs that are idempotent (i.e., jobs whose result and side effects are the same regardless of whether they are run once or many times).

max_job_retries : int, optional

Retry failed jobs this number of times before giving up. Even more than with `max_submit_retries`, this should only be used for jobs which are idempotent, as the job may have caused side effects (if any) before failing. These retries assist with jobs which may have failed because of network or worker failures.

hidden: bool, optional

The hidden status of the object. Setting this to true hides it from most API endpoints. The object can still be queried directly by ID. Defaults to True.

remote_backend : str or object, optional

The name of a joblib backend or a joblib backend itself. This parameter is the joblib backend to use when executing code within joblib in the container. The default of 'sequential' uses the joblib sequential backend in the container. The value 'civis' uses an exact copy of the Civis joblib backend that launched the container. Note that with the value 'civis', one can potentially use more jobs than specified by `n_jobs`.

****kwargs**:

Additional keyword arguments will be passed directly to `post_containers()`, potentially overriding the values of those arguments in the parent environment.

Raises **RuntimeError**

If this function is not running inside a Civis container job.

See also:

`civis.parallel.make_backend_factory`

```
civis.parallel.make_backend_factory(docker_image_name='civisanalytics/datascience-  
python', client=None, polling_interval=None,  
setup_cmd=None, max_submit_retries=0,  
max_job_retries=0, hidden=True, re-  
mote_backend='sequential', **kwargs)
```

Create a joblib backend factory that uses Civis Container Scripts

Note: The total size of function parameters in `Parallel()` calls on this backend must be less than 5 GB due to AWS file size limits.

Note: The maximum number of concurrent jobs in the Civis Platform is controlled by both the `n_jobs` and `pre_dispatch` parameters of `joblib.Parallel`. Set `pre_dispatch="n_jobs"` to have a maximum of `n_jobs` processes running at once. (The default is `pre_dispatch="2*n_jobs"`.)

Parameters `docker_image_name` : str, optional

The image for the container script. You may also wish to specify a `docker_image_tag` in the keyword arguments.

client : `civis.APIClient` instance or None, optional

An API Client object to use.

polling_interval : int, optional

The polling interval, in seconds, for checking container script status. If you have many jobs, you may want to set this higher (e.g., 300) to avoid *rate-limiting* <<https://platform.civisanalytics.com/api#basics>>. You should only set this if you aren't using pubnub notifications.

setup_cmd : str, optional

A shell command or sequence of commands for setting up the environment. These will precede the commands used to run functions in joblib. This is primarily for installing dependencies that are not available in the dockerhub repo (e.g., “cd /app && python setup.py install” or “pip install gensim”).

With no GitHub repo input, the setup command will default to a command that does nothing. If a `repo_http_uri` is provided, the default setup command will attempt to run “python setup.py install”. If this command fails, execution will still continue.

max_submit_retries : int, optional

The maximum number of retries for submitting each job. This is to help avoid a large set of jobs failing because of a single 5xx error. A value higher than zero should only be used for jobs that are idempotent (i.e., jobs whose result and side effects are the same regardless of whether they are run once or many times).

max_job_retries : int, optional

Retry failed jobs this number of times before giving up. Even more than with `max_submit_retries`, this should only be used for jobs which are idempotent, as the job may have caused side effects (if any) before failing. These retries assist with jobs which may have failed because of network or worker failures.

hidden: bool, optional

The hidden status of the object. Setting this to true hides it from most API endpoints. The object can still be queried directly by ID. Defaults to True.

remote_backend : str or object, optional

The name of a joblib backend or a joblib backend itself. This parameter is the joblib backend to use when executing code within joblib in the container. The default of 'sequential' uses the joblib sequential backend in the container. The value 'civis' uses an exact copy of the Civis joblib backend that launched the container. Note that with the value 'civis', one can potentially use more jobs than specified by `n_jobs`.

****kwargs:**

Additional keyword arguments will be passed directly to `post_containers()`.

See also:

`civis.APIClient.scripts.post_containers`

Notes

Joblib's `joblib.parallel.register_parallel_backend()` (see example above) expects a callable that returns a `joblib.parallel.ParallelBackendBase` instance. This function allows the user to specify the Civis container script setting that will be used when that backend creates container scripts to run jobs.

The specified Docker image (optionally, with a GitHub repo and setup command) must have basically the same environment as the one in which this module is used to submit jobs. The worker jobs need to be able to deserialize the jobs they are given, including the data and all the necessary Python objects (e.g., if you pass a Pandas data frame, the image must have Pandas installed). You may use functions and classes dynamically defined in the code (e.g. lambda functions), but if your joblib-parallized function calls code imported from another module, that module must be installed in the remote environment.

Examples

```
>>> # Without joblib:
>>> from __future__ import print_function
>>> from math import sqrt
>>> print([sqrt(i ** 2) for i in range(10)])
[0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0]
```

```
>>> # Using the default joblib backend:
>>> from joblib import delayed, Parallel
>>> parallel = Parallel(n_jobs=5)
>>> print(parallel(delayed(sqrt)(i ** 2) for i in range(10)))
[0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0]
```

```
>>> # Using the Civis backend:
>>> from joblib import parallel_backend, register_parallel_backend
>>> from civis.parallel import make_backend_factory
>>> register_parallel_backend('civis', make_backend_factory(
...     required_resources={"cpu": 512, "memory": 256}))
>>> with parallel_backend('civis'):
...     parallel = Parallel(n_jobs=5, pre_dispatch='n_jobs')
```

```
...     print(parallel(delayed(sqrt)(i ** 2) for i in range(10)))
[0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0]
```

```
>>> # Using scikit-learn with the Civis backend:
>>> from sklearn.externals.joblib import ...     register_parallel_backend as _
↳sklearn_register_parallel_backend
>>> from sklearn.externals.joblib import ...     parallel_backend as sklearn_
↳parallel_backend
>>> from sklearn.model_selection import GridSearchCV
>>> from sklearn.ensemble import GradientBoostingClassifier
>>> from sklearn.datasets import load_digits
>>> digits = load_digits()
>>> param_grid = {
...     "max_depth": [1, 3, 5, None],
...     "max_features": ["sqrt", "log2", None],
...     "learning_rate": [0.1, 0.01, 0.001]
... }
>>> # Note: n_jobs and pre_dispatch specify the maximum number of
>>> # concurrent jobs.
>>> gs = GridSearchCV(GradientBoostingClassifier(n_estimators=1000,
...                                             random_state=42),
...                   param_grid=param_grid,
...                   n_jobs=5, pre_dispatch="n_jobs")
>>> sklearn_register_parallel_backend('civis', make_backend_factory(
...     required_resources={"cpu": 512, "memory": 256}))
>>> with sklearn_parallel_backend('civis'):
...     gs.fit(digits.data, digits.target)
```

`civis.parallel.make_backend_template_factory` (*from_template_id*, *arguments=None*,
client=None, *polling_interval=None*,
max_submit_retries=0,
max_job_retries=0, *hidden=True*)

Create a joblib backend factory that uses Civis Custom Scripts.

Parameters from_template_id: int

Create jobs as Custom Scripts from the given template ID. When using the joblib backend with templates, the template must have a very specific form. Refer to the documentation for details.

arguments : dict or None, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params. See the *container scripts API documentation* <<https://platform.civisanalytics.com/api#resources-scripts>> for details.

client : *civis.APIClient* instance or None, optional

An API Client object to use.

polling_interval : int, optional

The polling interval, in seconds, for checking container script status. If you have many jobs, you may want to set this higher (e.g., 300) to avoid *rate-limiting* <<https://platform.civisanalytics.com/api#basics>>. You should only set this if you aren't using pubnub notifications.

max_submit_retries : int, optional

The maximum number of retries for submitting each job. This is to help avoid a large

set of jobs failing because of a single 5xx error. A value higher than zero should only be used for jobs that are idempotent (i.e., jobs whose result and side effects are the same regardless of whether they are run once or many times).

max_job_retries : int, optional

Retry failed jobs this number of times before giving up. Even more than with *max_submit_retries*, this should only be used for jobs which are idempotent, as the job may have caused side effects (if any) before failing. These retries assist with jobs which may have failed because of network or worker failures.

hidden: bool, optional

The hidden status of the object. Setting this to true hides it from most API endpoints. The object can still be queried directly by ID. Defaults to True.

5.5 API Client

APIClient is a class for handling requests to the Civis API. An instantiated *APIClient* contains a set of resources (listed below) where each resource is an object with methods. By convention, an instantiated *APIClient* object is named `client` and API requests are made with the following syntax:

```
client = civis.APIClient()
response = client.resource.method(params)
```

The methods on *APIClient* are created dynamically at runtime by parsing an `collections.OrderedDict` representation of the Civis API specification. By default, this specification is downloaded from the `/endpoints` endpoint the first time *APIClient* is instantiated (and cached in memory for the remainder of the program's run). In some circumstances, it may be useful to use a local cache of the API specification rather than downloading the spec. This can be done by passing the specification to the client through the parameter `local_api_spec` as either the `collections.OrderedDict` or a filename where the specification has been saved.

```
api_key = os.environ['CIVIS_API_KEY']
spec = civis.resources.get_api_spec(api_key)

# From OrderedDict
client = civis.APIClient(local_api_spec=spec)

# From file
with open('local_api_spec.json', 'w') as f:
    json.dump(spec, f)
client = civis.APIClient(local_api_spec='local_api_spec.json')
```

class `civis.APIClient` (*api_key=None*, *return_type='snake'*, *retry_total=6*, *api_version='1.0'*, *resources='base'*, *local_api_spec=None*)

The Civis API client.

Parameters `api_key` : str, optional

Your API key obtained from the Civis Platform. If not given, the client will use the `CIVIS_API_KEY` environment variable.

return_type : str, optional

The following types are implemented:

- 'raw' Returns the raw `requests.Response` object.

- 'snake' Returns a `civis.response.Response` object for the json-encoded content of a response. This maps the top-level json keys to snake_case.
- 'pandas' Returns a `pandas.DataFrame` for list-like responses and a `pandas.Series` for single a json response.

retry_total : int, optional

A number indicating the maximum number of retries for 429, 502, 503, or 504 errors.

api_version : string, optional

The version of endpoints to call. May instantiate multiple client objects with different versions. Currently only “1.0” is supported.

resources : string, optional

When set to “base”, only the default endpoints will be exposed in the client object. Set to “all” to include all endpoints available for a given user, including those that may be in development and subject to breaking changes at a later date.

local_api_spec : collections.OrderedDict or string, optional

The methods on this class are dynamically built from the Civis API specification, which can be retrieved from the /endpoints endpoint. When local_api_spec is None, the default, this specification is downloaded the first time APIClient is instantiated. Alternatively, a local cache of the specification may be passed as either an OrderedDict or a filename which points to a json file.

Attributes

announcements	An instance of the <i>Announcements</i> endpoint
apps	An instance of the <i>Apps</i> endpoint
clusters	An instance of the <i>Clusters</i> endpoint
codes	An instance of the <i>Codes</i> endpoint
credentials	An instance of the <i>Credentials</i> endpoint
databases	An instance of the <i>Databases</i> endpoint
endpoints	An instance of the <i>Endpoints</i> endpoint
enhancements	An instance of the <i>Enhancements</i> endpoint
exports	An instance of the <i>Exports</i> endpoint
files	An instance of the <i>Files</i> endpoint
groups	An instance of the <i>Groups</i> endpoint
imports	An instance of the <i>Imports</i> endpoint
jobs	An instance of the <i>Jobs</i> endpoint
match_targets	An instance of the <i>Match_Targets</i> endpoint
media	An instance of the <i>Media</i> endpoint
models	An instance of the <i>Models</i> endpoint
notebooks	An instance of the <i>Notebooks</i> endpoint
notifications	An instance of the <i>Notifications</i> endpoint
ontology	An instance of the <i>Ontology</i> endpoint
predictions	An instance of the <i>Predictions</i> endpoint
projects	An instance of the <i>Projects</i> endpoint
queries	An instance of the <i>Queries</i> endpoint
remote_hosts	An instance of the <i>Remote_Hosts</i> endpoint
reports	An instance of the <i>Reports</i> endpoint

Continued on next page

Table 5.4 – continued from previous page

results	An instance of the <i>Results</i> endpoint
scripts	An instance of the <i>Scripts</i> endpoint
search	An instance of the <i>Search</i> endpoint
tables	An instance of the <i>Tables</i> endpoint
templates	An instance of the <i>Templates</i> endpoint
users	An instance of the <i>Users</i> endpoint
workflows	An instance of the <i>Workflows</i> endpoint

default_credential

The current user's default credential.

get_aws_credential_id(*cred_name*, *owner=None*)

Find an AWS credential ID.

Parameters *cred_name* : str or int

If an integer ID is given, this passes through directly. If a str is given, return the ID corresponding to the AWS credential with that name.

owner : str, optional

Return the credential with this owner. If not provided, search for credentials under your username to disambiguate multiple credentials with the same name. Note that this function cannot return credentials which are not associated with an owner.

Returns *aws_credential_id* : int

The ID number of the AWS credentials.

Raises *ValueError*

If the AWS credential can't be found.

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.get_aws_credential_id('jsmith')
1234
```

```
>>> client.get_aws_credential_id(1111)
1111
```

```
>>> client.get_aws_credential_id('shared-cred',
...                             owner='research-group')
99
```

get_database_credential_id(*username*, *database_name*)

Return the credential ID for a given username in a given database.

Parameters *username* : str or int

If an integer ID is given, this passes through directly. If a str is given, return the ID corresponding to the database credential with that username.

database_name : str or int

Return the ID of the database credential with username *username* for this database name or ID.

Returns `database_credential_id` : int

The ID of the database credentials.

Raises `ValueError`

If the credential can't be found.

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.get_database_credential_id('jsmith', 'redshift-general')
1234
```

```
>>> client.get_database_credential_id(1111, 'redshift-general')
1111
```

get_database_id (*database*)

Return the database ID for a given database name.

Parameters `database` : str or int

If an integer ID is given, passes through. If a str is given the database ID corresponding to that database name is returned.

Returns `database_id` : int

The ID of the database.

Raises `ValueError`

If the database can't be found.

get_table_id (*table*, *database*)

Return the table ID for a given database and table name.

Parameters `table` : str

The name of the table in format schema.table.

database : str or int

The name or ID of the database.

Returns `table_id` : int

The ID of the table. Only returns exact match to specified table.

Raises `ValueError`

If an exact table match can't be found.

username

The current user's username.

5.5.1 API Response Types

class `civis.response.Response` (*json_data*, *snake_case=True*, *headers=None*)
Custom Civis response object.

Notes

The main features of this class are that it maps camelCase to snake_case at the top level of the json object and attaches keys as attributes. Nested object keys are not changed.

Attributes

<code>json_data</code>	(dict None) This is <i>json_data</i> as it is originally returned to the user without the key names being changed. See Notes. None is used if the original response returned a 204 No Content response.
<code>headers</code>	(dict) This is the header for the API call without changing the key names.
<code>calls_remaining</code>	(int) Number of API calls remaining before rate limit is reached.
<code>rate_limit</code>	(int) Total number of calls per API rate limit period.

class `civis.response.PaginatedResponse` (*path*, *initial_params*, *endpoint*)
A response object which is an iterator

Parameters *path* : str

Make GET requests to this path.

initial_params : dict

Query params that should be passed along with each request. Note that if *initial_params* contains the keys *page_num* or *limit*, they will be ignored. The given dict is not modified.

endpoint : *civis.base.Endpoint*

An endpoint used to make API requests.

Notes

This response is returned automatically by endpoints which support pagination when the *iterator* kwarg is specified.

Examples

```
>>> client = civis.APIClient()
>>> queries = client.queries.list(iterator=True)
>>> for query in queries:
...     print(query['id'])
```

class `civis.futures.CivisFuture` (*poller*, *poller_args*, *polling_interval=None*, *api_key=None*,
client=None, *poll_on_creation=True*)

A class for tracking future results.

This class will attempt to subscribe to a Pubnub channel to listen for job completion events. If you don't have access to Pubnub channels, then it will fallback to polling.

This is a subclass of `concurrent.futures.Future` from the Python standard library. See: <https://docs.python.org/3/library/concurrent.futures.html>

Parameters `poller` : func

A function which returns an object that has a `state` attribute.

`poller_args` : tuple

The arguments with which to call the poller function.

`polling_interval` : int or float, optional

The number of seconds between API requests to check whether a result is ready.

`api_key` : DEPRECATED str, optional

Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

`client` : `civis.APIClient`, optional

`poll_on_creation` : bool, optional

If `True` (the default), it will poll upon calling `result()` the first time. If `False`, it will wait the number of seconds specified in `polling_interval` from object creation before polling.

Examples

This example is provided as a function at `query_civis()`.

```
>>> client = civis.APIClient()
>>> database_id = client.get_database_id("my_database")
>>> cred_id = client.default_credential
>>> sql = "SELECT 1"
>>> preview_rows = 10
>>> response = client.queries.post(database_id, sql, preview_rows,
>>>                                credential=cred_id)
>>> job_id = response.id
>>>
>>> poller = client.queries.get
>>> poller_args = (job_id, ) # (job_id, run_id) if poller requires run_id
>>> polling_interval = 10
>>> future = CivisFuture(poller, poller_args, polling_interval)
```

5.5.2 API Resources

Announcements

class `Announcements` (*session_kwargs*, *return_type*='civis')

Methods

<code>list(*[, limit, page_num, order, order_dir, ...])</code>	List announcements
--	--------------------

list (*, limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', iterator='DEFAULT')

List announcements

Parameters **limit** : integer, optional

Number of results to return. Defaults to 10. Maximum allowed is 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to released_at. Must be one of: released_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The ID of this announcement

subject : string

The subject of this announcement.

body : string

The body of this announcement.

released_at : string/date-time

The date and time this announcement was released.

created_at : string/date-time

updated_at : string/date-time

Apps

class Apps (session_kwargs, return_type='civis')

Methods

<code>delete_instances_projects(id, slug)</code>	<code>project_id,</code>	Remove a AppInstance from a project
<code>delete_instances_shares_groups(id, group_id, ...)</code>		Revoke the permissions a group has on this object
<code>delete_instances_shares_users(id, slug)</code>	<code>user_id,</code>	Revoke the permissions a user has on this object
<code>get(slug)</code>		List details of a Decision Application

Continued on next page

Table 5.6 – continued from previous page

<code>get_instances(id, slug)</code>	Return a given app instance
<code>list()</code>	List apps
<code>list_instances(slug, *[, archived, ...])</code>	List the instances of a Decision Application
<code>list_instances_projects(id, slug, *[, hidden])</code>	List the projects a AppInstance belongs to
<code>list_instances_shares(id, slug)</code>	List users and groups permissioned on this object
<code>patch_instances(id, slug, *[, name])</code>	Update a given app instance
<code>post_instances(slug, *[, name])</code>	Create a new instance of an application of the given slug
<code>put_instances_archive(id, slug, status)</code>	Update the archive status of this object
<code>put_instances_projects(id, project_id, slug)</code>	Add a AppInstance to a project
<code>put_instances_shares_groups(id, slug, ...[, ...])</code>	Set the permissions groups has on this object
<code>put_instances_shares_users(id, slug, ...[, ...])</code>	Set the permissions users have on this object

`delete_instances_projects` (*id, project_id, slug*)

Remove a AppInstance from a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

slug : string

The slug for the application.

Returns None

Response code 204: success

`delete_instances_shares_groups` (*id, group_id, slug*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

slug : string

The slug for the application.

Returns None

Response code 204: success

`delete_instances_shares_users` (*id, user_id, slug*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

slug : string

The slug for the application.

Returns None

Response code 204: success

get (*slug*)

List details of a Decision Application

Parameters **slug** : string

The slug for the application.

Returns **slug** : string

The slug for the application.

id : integer

The unique id of the application.

instance_name : string

A word that describes an instance of this app.

name : string

The name of the application.

current_release : dict:

```
- id : integer
    The unique id of the release.
- app_id : integer
    The id of the app the release belongs to.
- report_template_id : integer
    ID of the report template for this release.
- resources : dict
    A hash of resources associated with this release.
```

features : dict

App features.

get_instances (*id*, *slug*)

Return a given app instance

Parameters **id** : integer

The unique id of the instance.

slug : string

The slug for the application.

Returns **id** : integer

The unique id of the instance.

name : string

The name of the instance.

app_release_id : integer

The id of the app release the instance belongs to.

report_id : integer

The id of the report the instance belongs to.

created_at : string/time

The time the instance was created at.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

project_id : integer

The id of the project collecting all the objects that belong to this app instance.

auth_code_url : string

api_key : string

A Civis API key that can be used by this app instance.

list()

List apps

Returns slug : string

The slug for the application.

id : integer

The unique id of the application.

instance_name : string

A word that describes an instance of this app.

name : string

The name of the application.

list_instances (*slug*, *, *archived*='DEFAULT', *app_release_id*='DEFAULT', *limit*='DEFAULT',
 page_num='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List the instances of a Decision Application

Parameters slug : string

The slug for the application.

archived : string, optional

The archival status of the requested object(s).

app_release_id : integer, optional

If supplied, return only instances matching this release.

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id, created_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to asc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns id : integer

The unique id of the instance.

name : string

The name of the instance.

app_release_id : integer

The id of the app release the instance belongs to.

report_id : integer

The id of the report the instance belongs to.

created_at : string/time

The time the instance was created at.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

project_id : integer

The id of the project collecting all the objects that belong to this app instance.

list_instances_projects (*id*, *slug*, *, *hidden*=*'DEFAULT'*)

List the projects a AppInstance belongs to

Parameters id : integer

The ID of the resource.

slug : string

The slug for the application.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns `id` : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_instances_shares (*id*, *slug*)

List users and groups permissioned on this object

Parameters `id` : integer

The ID of the object.

slug : string

The slug for the application.

Returns `readers` : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch_instances (*id*, *slug*, *, *name*='DEFAULT')

Update a given app instance

Parameters *id* : integer

The unique id of the instance.

slug : string

The slug for the application.

name : string, optional

The name of the instance.

Returns *id* : integer

The unique id of the instance.

name : string

The name of the instance.

app_release_id : integer

The id of the app release the instance belongs to.

report_id : integer

The id of the report the instance belongs to.

created_at : string/time

The time the instance was created at.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

project_id : integer

The id of the project collecting all the objects that belong to this app instance.

auth_code_url : string

api_key : string

A Civis API key that can be used by this app instance.

post_instances (*slug*, *, *name*='DEFAULT')

Create a new instance of an application of the given slug

Parameters **slug** : string

The slug for the application.

name : string, optional

The name of the instance.

Returns **id** : integer

The unique id of the instance.

name : string

The name of the instance.

app_release_id : integer

The id of the app release the instance belongs to.

report_id : integer

The id of the report the instance belongs to.

created_at : string/time

The time the instance was created at.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
```

```

    This user's username.
-   initials : string
    This user's initials.
-   online : boolean
    Whether this user is online.

```

project_id : integer

The id of the project collecting all the objects that belong to this app instance.

auth_code_url : string

api_key : string

A Civis API key that can be used by this app instance.

put_instances_archive (*id*, *slug*, *status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

slug : string

The slug for the application.

status : boolean

The desired archived status of the object.

Returns **id** : integer

The unique id of the instance.

name : string

The name of the instance.

app_release_id : integer

The id of the app release the instance belongs to.

report_id : integer

The id of the report the instance belongs to.

created_at : string/time

The time the instance was created at.

user : dict:

```

-   id : integer
    The ID of this user.
-   name : string
    This user's name.
-   username : string
    This user's username.
-   initials : string
    This user's initials.
-   online : boolean
    Whether this user is online.

```

project_id : integer

The id of the project collecting all the objects that belong to this app instance.

auth_code_url : string

api_key : string

A Civis API key that can be used by this app instance.

put_instances_projects (*id, project_id, slug*)

Add a AppInstance to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

slug : string

The slug for the application.

Returns None

Response code 204: success

put_instances_shares_groups (*id, slug, group_ids, permission_level, *, share_email_body='DEFAULT', send_shared_email='DEFAULT'*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

slug : string

The slug for the application.

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_instances_shares_users (*id*, *slug*, *user_ids*, *permission_level*,
*, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

slug : string

The slug for the application.

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns *readers* : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Clusters

class Clusters (*session_kwargs*, *return_type*='civis')

Methods

<code>get_kubernetes(id)</code>	Describe a Kubernetes Cluster
<code>get_workers(id)</code>	Describe a Worker Cluster
<code>list_kubernetes(*[, organization_slug, ...])</code>	List Kubernetes Clusters
<code>list_workers(*[, limit, page_num, order, ...])</code>	List Worker Clusters
<code>list_workers_active_jobs(id)</code>	List Active Jobs for a Worker Cluster
<code>list_workers_queued_jobs(id)</code>	List Queued Jobs for a Worker Cluster

get_kubernetes (*id*)

Describe a Kubernetes Cluster

Parameters **id** : integer

The ID of this cluster.

Returns **id** : integer

The ID of this cluster.

instance_type : string

The EC2 instance types in this cluster.

min_instances : integer

The minimum number of instances in this cluster.

max_instances : integer

The maximum number of instances in this cluster.

instance_max_memory : integer

The amount of memory available to a single instance.

instance_max_cpu : integer

The number of processor shares available to a single instance.

organization_id : string

The id of this cluster's organization.

organization_slug : string

The slug of this cluster's organization.

security_group_id : string

The security group to be added to the nodes of this cluster

get_workers (*id*)

Describe a Worker Cluster

Parameters id : integer

The ID of this cluster.

Returns id : integer

The ID of this cluster.

instance_type : string

The EC2 instance types in this cluster.

min_instances : integer

The minimum number of instances in this cluster.

max_instances : integer

The maximum number of instances in this cluster.

instances : integer

The number of instances currently in this cluster.

instance_max_memory : integer

The amount of memory available to a single instance.

instance_max_cpu : integer

The number of processor shares available to a single instance.

instance_max_disk_space : number/float

The amount of memory available to a single instance.

region : string

The AWS region that this cluster is in.

active_jobs_count : integer

The number of jobs currently being run in the cluster.

queued_jobs_count : integer

The number of jobs currently waiting to be run on the cluster.

list_kubernetes (*, *organization_slug*='DEFAULT', *limit*='DEFAULT', *page_num*='DEFAULT',
order='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List Kubernetes Clusters

Parameters **organization_slug** : string, optional

The slug of this cluster's organization.

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to *organization_id*. Must be one of: *organization_id*, *created_at*.

order_dir : string, optional

Direction in which to sort, either *asc* (ascending) or *desc* (descending) defaulting to *asc*.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by *limit* are needed. When True, *limit* and *page_num* are ignored. Defaults to False.

Returns **id** : integer

The ID of this cluster.

instance_type : string

The EC2 instance types in this cluster.

min_instances : integer

The minimum number of instances in this cluster.

max_instances : integer

The maximum number of instances in this cluster.

instance_max_memory : integer

The amount of memory available to a single instance.

instance_max_cpu : integer

The number of processor shares available to a single instance.

organization_id : string

The id of this cluster's organization.

organization_slug : string

The slug of this cluster's organization.

security_group_id : string

The security group to be added to the nodes of this cluster

```
list_workers (*, limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', or-
               der_dir='DEFAULT', iterator='DEFAULT')
```

List Worker Clusters

Parameters **limit** : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id, created_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to asc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The ID of this cluster.

instance_type : string

The EC2 instance types in this cluster.

min_instances : integer

The minimum number of instances in this cluster.

max_instances : integer

The maximum number of instances in this cluster.

region : string

The AWS region that this cluster is in.

active_jobs_count : integer

The number of jobs currently being run in the cluster.

queued_jobs_count : integer

The number of jobs currently waiting to be run on the cluster.

```
list_workers_active_jobs (id)
```

List Active Jobs for a Worker Cluster

Parameters **id** : integer

The ID of this cluster.

Returns **id** : integer

name : string

type : string

state : string

Whether the job is idle, queued, running, cancelled, or failed.

created_at : string/date-time

updated_at : string/date-time

runs : list:

Information about the most recent runs of the job.

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

required_cpu : integer

The CPU shares required by the script.

required_disk_space : integer

The disk space in GB required by the script.

required_memory : integer

The memory in MB required by the script.

author : dict:

```
- id : integer
  The ID of this user.
- name : string
  This user's name.
- username : string
  This user's username.
- initials : string
  This user's initials.
- online : boolean
  Whether this user is online.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

list_workers_queued_jobs (*id*)

List Queued Jobs for a Worker Cluster

Parameters *id* : integer

The ID of this cluster.

Returns *id* : integer

name : string

type : string

state : string

Whether the job is idle, queued, running, cancelled, or failed.

created_at : string/date-time

updated_at : string/date-time

runs : list:

```
Information about the most recent runs of the job.
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

required_cpu : integer

The CPU shares required by the script.

required_disk_space : integer

The disk space in GB required by the script.

required_memory : integer

The memory in MB required by the script.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

Codes

class **Codes** (*session_kwargs*, *return_type*='civis')

Methods

<code>delete(id)</code>	Delete a code
<code>get(id)</code>	Show basic code info
<code>list(*[, limit, type, featured])</code>	List codes
<code>patch(id, *[, name, type, body, readme, ...])</code>	Update a code
<code>post(name, type, body, readme, *[, score, ...])</code>	Create a new code
<code>put(id, name, type, body, readme, *[, ...])</code>	Update a code

delete (*id*)
Delete a code

Parameters **id** : integer

The ID for this code.

Returns None

Response code 204: success

get (*id*)

Show basic code info

Parameters **id** : integer

The ID for this code.

Returns **id** : integer

The ID for this code.

name : string

Name of code.

type : string

The code's type; e.g., Code::FrontEnd.

body : string

Actual code contents; e.g., HTML, SQL, etc.

readme : string

User specified information about this code. Markdown format accepted.

score : integer

Internal Civis Use Only.

auth_s3_url : string

Authorized_s3_link to file.

created_at : string/time

The creation time for this code.

updated_at : string/time

The last modification time for this code.

list (*, *limit*=*'DEFAULT'*, *type*=*'DEFAULT'*, *featured*=*'DEFAULT'*)

List codes

Parameters **limit** : integer, optional

The maximum number of codes to return.

type : string, optional

The code's type; e.g., Code::FrontEnd.

featured : boolean, optional

If true, return featured codes.

Returns **id** : integer

The ID for this code.

name : string

Name of code.

type : string

The code's type; e.g., Code::FrontEnd.

created_at : string/time

The creation time for this code.

updated_at : string/time

The last modification time for this code.

patch (*id*, *, *name*='DEFAULT', *type*='DEFAULT', *body*='DEFAULT', *readme*='DEFAULT',
score='DEFAULT', *auth_s3_url*='DEFAULT')
Update a code

Parameters *id* : integer

The ID for this code.

name : string, optional

Name of code.

type : string, optional

The code's type; e.g., Code::FrontEnd.

body : string, optional

Actual code contents; e.g., HTML, SQL, etc.

readme : string, optional

User specified information about this code. Markdown format accepted.

score : integer, optional

Internal Civis Use Only.

auth_s3_url : string, optional

Authorized_s3_link to file.

Returns None

Response code 204: success

post (*name*, *type*, *body*, *readme*, *, *score*='DEFAULT', *auth_s3_url*='DEFAULT')
Create a new code

Parameters *name* : string

Name of code.

type : string

The code's type; e.g., Code::FrontEnd.

body : string

Actual code contents; e.g., HTML, SQL, etc.

readme : string

User specified information about this code. Markdown format accepted.

score : integer, optional

Internal Civis Use Only.

auth_s3_url : string, optional

Authorized_s3_link to file.

Returns None

Response code 204: success

put (*id, name, type, body, readme, *, score='DEFAULT', auth_s3_url='DEFAULT'*)
Update a code

Parameters **id** : integer

The ID for this code.

name : string

Name of code.

type : string

The code's type; e.g., Code::FrontEnd.

body : string

Actual code contents; e.g., HTML, SQL, etc.

readme : string

User specified information about this code. Markdown format accepted.

score : integer, optional

Internal Civis Use Only.

auth_s3_url : string, optional

Authorized_s3_link to file.

Returns **id** : integer

The ID for this code.

name : string

Name of code.

type : string

The code's type; e.g., Code::FrontEnd.

body : string

Actual code contents; e.g., HTML, SQL, etc.

readme : string

User specified information about this code. Markdown format accepted.

score : integer

Internal Civis Use Only.

auth_s3_url : string

Authorized_s3_link to file.

created_at : string/time

The creation time for this code.

updated_at : string/time

The last modification time for this code.

Credentials

class Credentials (*session_kwargs*, *return_type*='civis')

Methods

<i>delete_shares_groups</i> (id, group_id)	Revoke the permissions a group has on this object
<i>delete_shares_users</i> (id, user_id)	Revoke the permissions a user has on this object
<i>get</i> (id)	Get a credential
<i>list</i> (*[, type, default, limit, page_num, ...])	List credentials
<i>list_shares</i> (id)	List users and groups permissioned on this object
<i>post</i> (type, username, password, *[, name, ...])	Create a credential
<i>post_authenticate</i> (url, remote_host_type, ...)	Authenticate against a remote host
<i>post_temporary</i> (id, *[, duration])	Generate a temporary credential for accessing S3
<i>put</i> (id, type, username, password, *[, name, ...])	Update an existing credential
<i>put_shares_groups</i> (id, group_ids, ...[, ...])	Set the permissions groups has on this object
<i>put_shares_users</i> (id, user_ids, ...[, ...])	Set the permissions users have on this object

delete_shares_groups (*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users (*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get (*id*)

Get a credential

Parameters **id** : integer

The ID of the credential.

Returns id : integer

The ID of the credential.

name : string

The name identifying the credential

type : string

The credential's type.

username : string

The username for the credential.

description : string

A long description of the credential.

owner : string

The name of the user who this credential belongs to.

remote_host_id : integer

The ID of the remote host associated with this credential.

remote_host_name : string

The name of the remote host associated with this credential.

created_at : string/time

The creation time for this credential.

updated_at : string/time

The last modification time for this credential.

list (*, type='DEFAULT', default='DEFAULT', limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', iterator='DEFAULT')
List credentials

Parameters type : string, optional

The type (or types) of credentials to return. One or more of: Amazon Web Services S3, BSD::API, CASS/NCOA PAF, Catalist::API, Catalist::SFTP, Certificate, Civis Platform, Custom, Database, Google, Github, JobTraits::Ftp, Salesforce User, Salesforce Client, Silverpop Application, Silverpop Refresh Token, Silverpop User, TableauUser, VAN::MyVoterFile, VAN::MyCampaign, and VAN::BothModes. Specify multiple values as a comma-separated list (e.g., "A,B").

default : boolean, optional

If true, will return a list with a single credential which is the current user's default credential.

limit : integer, optional

Number of results to return. Defaults to its maximum of 1000.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to `updated_at`. Must be one of: `updated_at`, `created_at`, `name`.

order_dir : string, optional

Direction in which to sort, either `asc` (ascending) or `desc` (descending) defaulting to `desc`.

iterator : bool, optional

If `True`, return a generator to iterate over all responses. Use when more results than the maximum allowed by `limit` are needed. When `True`, `limit` and `page_num` are ignored. Defaults to `False`.

Returns **id** : integer

The ID of the credential.

name : string

The name identifying the credential

type : string

The credential's type.

username : string

The username for the credential.

description : string

A long description of the credential.

owner : string

The name of the user who this credential belongs to.

remote_host_id : integer

The ID of the remote host associated with this credential.

remote_host_name : string

The name of the remote host associated with this credential.

created_at : string/time

The creation time for this credential.

updated_at : string/time

The last modification time for this credential.

list_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
```

```
- id : integer
- name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

post (*type*, *username*, *password*, *, *name*='DEFAULT', *description*='DEFAULT', *remote_host_id*='DEFAULT', *system_credential*='DEFAULT')

Create a credential

Parameters *type* : string

username : string

The username for the credential.

password : string

The password for the credential.

name : string, optional

The name identifying the credential.

description : string, optional

A long description of the credential.

remote_host_id : integer, optional

The ID of the remote host associated with the credential.

system_credential : boolean, optional

Returns *id* : integer

The ID of the credential.

name : string

The name identifying the credential

type : string

The credential's type.

username : string

The username for the credential.

description : string

A long description of the credential.

owner : string

The name of the user who this credential belongs to.

remote_host_id : integer

The ID of the remote host associated with this credential.

remote_host_name : string

The name of the remote host associated with this credential.

created_at : string/time

The creation time for this credential.

updated_at : string/time

The last modification time for this credential.

post_authenticate (*url, remote_host_type, username, password*)

Authenticate against a remote host

Parameters **url** : string

The URL to your host.

remote_host_type : string

The type of remote host. One of: RemoteHostTypes::BSD, RemoteHostTypes::Ftp, RemoteHostTypes::Github, RemoteHostTypes::GoogleDoc, RemoteHostTypes::JDBC, RemoteHostTypes::Redshift, RemoteHostTypes::Salesforce, and RemoteHostTypes::Van

username : string

The username for the credential.

password : string

The password for the credential.

Returns **id** : integer

The ID of the credential.

name : string

The name identifying the credential

type : string

The credential's type.

username : string

The username for the credential.

description : string

A long description of the credential.

owner : string

The name of the user who this credential belongs to.

remote_host_id : integer

The ID of the remote host associated with this credential.

remote_host_name : string

The name of the remote host associated with this credential.

created_at : string/time

The creation time for this credential.

updated_at : string/time

The last modification time for this credential.

post_temporary (*id*, *, *duration*=*'DEFAULT'*)

Generate a temporary credential for accessing S3

Parameters *id* : integer

The ID of the credential.

duration : integer, optional

The number of seconds the temporary credential should be valid. Defaults to 15 minutes. Must not be less than 15 minutes or greater than 36 hours.

Returns *access_key* : string

The identifier of the credential.

secret_access_key : string

The secret part of the credential.

session_token : string

The session token identifier.

put (*id*, *type*, *username*, *password*, *, *name*=*'DEFAULT'*, *description*=*'DEFAULT'*, *remote_host_id*=*'DEFAULT'*, *system_credential*=*'DEFAULT'*)

Update an existing credential

Parameters *id* : integer

The ID of the credential.

type : string

username : string

The username for the credential.

password : string

The password for the credential.

name : string, optional

The name identifying the credential.

description : string, optional

A long description of the credential.

remote_host_id : integer, optional

The ID of the remote host associated with the credential.

system_credential : boolean, optional

Returns id : integer

The ID of the credential.

name : string

The name identifying the credential

type : string

The credential's type.

username : string

The username for the credential.

description : string

A long description of the credential.

owner : string

The name of the user who this credential belongs to.

remote_host_id : integer

The ID of the remote host associated with this credential.

remote_host_name : string

The name of the remote host associated with this credential.

created_at : string/time

The creation time for this credential.

updated_at : string/time

The last modification time for this credential.

put_shares_groups (*id*, *group_ids*, *permission_level*, *, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions groups has on this object

Parameters id : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: "read", "write", or "manage"

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns `readers` : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns `readers` : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Databases

class Databases (*session_kwargs*, *return_type*='civis')

Methods

<code>delete_whitelist_ips(id, whitelisted_ip_id)</code>	Remove a whitelisted IP address
<code>get_whitelist_ips(id, whitelisted_ip_id)</code>	View details about a whitelisted IP
<code>list()</code>	List databases
<code>list_schemas(id)</code>	List schemas in this database
<code>list_whitelist_ips(id)</code>	List whitelisted IPs for the specified database
<code>post_whitelist_ips(id, subnet_mask)</code>	Whitelist an IP address

delete_whitelist_ips (*id*, *whitelisted_ip_id*)

Remove a whitelisted IP address

Parameters **id** : integer

The ID of the database this rule is applied to.

whitelisted_ip_id : integer

The ID of this whitelisted IP address.

Returns None

Response code 204: success

get_whitelist_ips (*id*, *whitelisted_ip_id*)

View details about a whitelisted IP

Parameters **id** : integer

The ID of the database this rule is applied to.

whitelisted_ip_id : integer

The ID of this whitelisted IP address.

Returns **id** : integer

The ID of this whitelisted IP address.

remote_host_id : integer

The ID of the database this rule is applied to.

security_group_id : string

The ID of the security group this rule is applied to.

subnet_mask : string

The subnet mask that is allowed by this rule.

authorized_by : string

The user who authorized this rule.

is_active : boolean

True if the rule is applied, false if it has been revoked.

created_at : string/time

The time this rule was created.

updated_at : string/time

The time this rule was last updated.

list ()

List databases

Returns **id** : integer

The ID for the database.

name : string

The name of the database.

list_schemas (*id*)

List schemas in this database

Parameters **id** : integer

The ID of the database.

Returns **schema** : string

The name of a schema.

list_whitelist_ips (*id*)

List whitelisted IPs for the specified database

Parameters **id** : integer

The ID for the database.

Returns **id** : integer

The ID of this whitelisted IP address.

remote_host_id : integer

The ID of the database this rule is applied to.

security_group_id : string

The ID of the security group this rule is applied to.

subnet_mask : string

The subnet mask that is allowed by this rule.

created_at : string/time

The time this rule was created.

updated_at : string/time

The time this rule was last updated.

post_whitelist_ips (*id*, *subnet_mask*)

Whitelist an IP address

Parameters **id** : integer

The ID of the database this rule is applied to.

subnet_mask : string

The subnet mask that is allowed by this rule.

Returns **id** : integer

The ID of this whitelisted IP address.

remote_host_id : integer

The ID of the database this rule is applied to.

security_group_id : string

The ID of the security group this rule is applied to.

subnet_mask : string

The subnet mask that is allowed by this rule.

authorized_by : string

The user who authorized this rule.

is_active : boolean

True if the rule is applied, false if it has been revoked.

created_at : string/time

The time this rule was created.

updated_at : string/time

The time this rule was last updated.

Endpoints

class Endpoints (*session_kwargs*, *return_type*='civis')

Methods

<i>list()</i>	List API endpoints
list()	List API endpoints
Returns	None
	Response code 200: success

Enhancements

class Enhancements (*session_kwargs*, *return_type*='civis')

Methods

<i>delete_cass_ncoa_projects</i> (id, project_id)	Remove a JobTypes::CassNcoa from a project
<i>delete_cass_ncoa_runs</i> (id, run_id)	Cancel a run
<i>delete_cass_ncoa_shares_groups</i> (id, group_id)	Revoke the permissions a group has on this object
<i>delete_cass_ncoa_shares_users</i> (id, user_id)	Revoke the permissions a user has on this object
<i>get_cass_ncoa</i> (id)	Get a CASS/NCOA Enhancement
<i>get_cass_ncoa_runs</i> (id, run_id)	Check status of a run
<i>list</i> (*[, type, author, status, archived, ...])	List Enhancements
<i>list_cass_ncoa_projects</i> (id, *[, hidden])	List the projects a JobTypes::CassNcoa belongs to
<i>list_cass_ncoa_runs</i> (id, *[, limit, ...])	List runs for the given cass_ncoa
<i>list_cass_ncoa_runs_logs</i> (id, run_id, *[, ...])	Get the logs for a run
<i>list_cass_ncoa_runs_outputs</i> (id, run_id, *[, ...])	List the outputs for a run
<i>list_cass_ncoa_shares</i> (id)	List users and groups permissioned on this object
<i>list_types</i> ()	List available enhancement types
<i>patch_cass_ncoa</i> (id, *[, name, schedule, ...])	Update some attributes of this CASS/NCOA Enhancement
<i>post_cass_ncoa</i> (name, source, *[, schedule, ...])	Create a CASS/NCOA Enhancement
<i>post_cass_ncoa_cancel</i> (id)	Cancel a run
<i>post_cass_ncoa_runs</i> (id)	Start a run
<i>put_cass_ncoa</i> (id, name, source, *[, ...])	Replace all attributes of this CASS/NCOA Enhancement
<i>put_cass_ncoa_archive</i> (id, status)	Update the archive status of this object

Continued on next page

Table 5.12 – continued from previous page

<code>put_cass_ncoa_projects(id, project_id)</code>	Add a JobTypes::CassNcoa to a project
<code>put_cass_ncoa_shares_groups(id, group_ids, ...)</code>	Set the permissions groups has on this object
<code>put_cass_ncoa_shares_users(id, user_ids, ...)</code>	Set the permissions users have on this object

delete_cass_ncoa_projects (*id, project_id*)

Remove a JobTypes::CassNcoa from a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_cass_ncoa_runs (*id, run_id*)

Cancel a run

Parameters *id* : integer

The ID of the cass_ncoa.

run_id : integer

The ID of the run.

Returns None

Response code 202: success

delete_cass_ncoa_shares_groups (*id, group_id*)

Revoke the permissions a group has on this object

Parameters *id* : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_cass_ncoa_shares_users (*id, user_id*)

Revoke the permissions a user has on this object

Parameters *id* : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get_cass_ncoa (*id*)

Get a CASS/NCOA Enhancement

Parameters *id* : integer**Returns** *id* : integer

The ID for the enhancement.

name : string

The name of the enhancement.

type : string

The type of the enhancement (e.g CASS-NCOA)

created_at : string/time

The time this enhancement was created.

updated_at : string/time

The time the enhancement was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the enhancement's last run

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

parent_id : integer

Parent ID that triggers this enhancement.

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
↳ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
↳ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

source : dict:

```
- database_table : dict::
    - schema : string
        The schema name of the source table.
    - table : string
        The name of the source table.
    - remote_host_id : integer
        The ID of the database host for the table.
    - credential_id : integer
        The id of the credentials to be used when performing the
        enhancement.
    - multipart_key : list
        The source table primary key.
```

destination : dict:

```
- database_table : dict::
    - schema : string
        The schema name for the output data.
    - table : string
```


The table name **for** the output data.

column_mapping : dict:

```
- address1 : string
    The first address line.
- address2 : string
    The second address line.
- city : string
    The city of an address.
- state : string
    The state of an address.
- zip : string
    The zip code of an address.
- name : string
    The full name of the resident at this address. If needed, ↵
↵separate
    multiple columns with `+`, e.g. `first_name+last_name`
- company : string
    The name of the company located at this address.
```

use_default_column_mapping : boolean

Defaults to true, where the existing column mapping on the input table will be used. If false, a custom column mapping must be provided.

perform_ncoa : boolean

Whether to update addresses for records matching the National Change of Address (NCOA) database.

ncoa_credential_id : integer

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

output_level : string

The set of fields persisted by a CASS or NCOA enhancement. For CASS enhancements, one of 'cass' or 'all.' For NCOA enhancements, one of 'cass', 'ncoa', 'coalesced' or 'all'. By default, all fields will be returned.

limiting_sql : string

The limiting sql.

archived : string

The archival status of the requested object(s).

get_cass_ncoa_runs (*id*, *run_id*)

Check status of a run

Parameters *id* : integer

The ID of the cass_ncoa.

run_id : integer

The ID of the run.

Returns *id* : integer

The ID of the run.

cass_ncoa_id : integer

The ID of the cass_ncoa.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

list (*, type='DEFAULT', author='DEFAULT', status='DEFAULT', archived='DEFAULT', limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', iterator='DEFAULT')

List Enhancements

Parameters **type** : string, optional

If specified, return objects of these types.

author : string, optional

If specified, return objects from this author. Must use user IDs. A comma separated list of IDs is also accepted to return objects from multiple authors.

status : string, optional

If specified, returns objects with one of these statuses. It accepts a comma-separated list, possible values are 'running', 'failed', 'succeeded', 'idle', 'scheduled'.

archived : string, optional

The archival status of the requested object(s).

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to updated_at. Must be one of: updated_at, name, created_at, last_run.updated_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The ID for the enhancement.

name : string

The name of the enhancement.

type : string

The type of the enhancement (e.g CASS-NCOA)

created_at : string/time

The time this enhancement was created.

updated_at : string/time

The time the enhancement was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the enhancement's last run

archived : string

The archival status of the requested object(s).

list_cass_ncoa_projects (*id*, *, *hidden*='DEFAULT')

List the projects a JobTypes::CassNcoa belongs to

Parameters **id** : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns **id** : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
```

```
This user's name.
- username : string
  This user's username.
- initials : string
  This user's initials.
- online : boolean
  Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
  The ID of this user.
- name : string
  This user's name.
- username : string
  This user's username.
- initials : string
  This user's initials.
- online : boolean
  Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_cass_ncoa_runs (*id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List runs for the given cass_ncoa

Parameters **id** : integer

The ID of the cass_ncoa.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 100.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns id : integer

The ID of the run.

cass_ncoa_id : integer

The ID of the cass_ncoa.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

list_cass_ncoa_runs_logs (*id*, *run_id*, *, *last_id*=*'DEFAULT'*, *limit*=*'DEFAULT'*)

Get the logs for a run

Parameters id : integer

The ID of the cass_ncoa.

run_id : integer

The ID of the run.

last_id : integer, optional

The ID of the last log message received. Log entries with this ID value or lower will be omitted. Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.

limit : integer, optional

The maximum number of log messages to return. Default of 10000.

Returns id : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

message : string

The log message.

level : string

The level of the log. One of unknown,fatal,error,warn,info,debug.

```
list_cass_ncoa_runs_outputs (id, run_id, *, limit='DEFAULT', page_num='DEFAULT',  
                                order='DEFAULT', order_dir='DEFAULT', itera-  
                                tor='DEFAULT')
```

List the outputs for a run

Parameters *id* : integer

The ID of the job.

run_id : integer

The ID of the run.

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to `created_at`. Must be one of: `created_at`, `id`.

order_dir : string, optional

Direction in which to sort, either `asc` (ascending) or `desc` (descending) defaulting to `desc`.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by `limit` are needed. When True, `limit` and `page_num` are ignored. Defaults to False.

Returns *object_type* : string

The type of the output. Valid values are `File`, `Table`, `Report`, `Project`, or `Credential`

object_id : integer

The ID of the output object.

name : string

The name of the output object.

link : string

The link to retrieve the output object.

```
list_cass_ncoa_shares (id)
```

List users and groups permissioned on this object

Parameters *id* : integer

The ID of the object.

Returns *readers* : dict:

```
- users : list::  
  - id : integer  
  - name : string  
- groups : list::  
  - id : integer  
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_types ()

List available enhancement types

Returns name : string

The name of the type.

patch_cass_ncoa (id, *, name='DEFAULT', schedule='DEFAULT', parent_id='DEFAULT', notifications='DEFAULT', source='DEFAULT', destination='DEFAULT', column_mapping='DEFAULT', use_default_column_mapping='DEFAULT', perform_ncoa='DEFAULT', ncoa_credential_id='DEFAULT', output_level='DEFAULT', limiting_sql='DEFAULT')

Update some attributes of this CASS/NCOA Enhancement

Parameters id : integer

The ID for the enhancement.

name : string, optional

The name of the enhancement.

schedule : dict, optional:

```
- scheduled : boolean
  If the object is scheduled
- scheduled_days : list
  Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
  Hours of the day it is scheduled on
- scheduled_minutes : list
  Minutes of the day it is scheduled on
```

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

parent_id : integer, optional

Parent ID that triggers this enhancement.

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

source : dict, optional:

```
- database_table : dict::
    - schema : string
        The schema name of the source table.
    - table : string
        The name of the source table.
    - remote_host_id : integer
        The ID of the database host for the table.
    - credential_id : integer
        The id of the credentials to be used when performing the
        enhancement.
    - multipart_key : list
        The source table primary key.
```

destination : dict, optional:

```
- database_table : dict::
    - schema : string
        The schema name for the output data.
    - table : string
        The table name for the output data.
```

column_mapping : dict, optional:


```

- address1 : string
    The first address line.
- address2 : string
    The second address line.
- city : string
    The city of an address.
- state : string
    The state of an address.
- zip : string
    The zip code of an address.
- name : string
    The full name of the resident at this address. If needed, ↵
↵separate
    multiple columns with `+`, e.g. `first_name+last_name`
- company : string
    The name of the company located at this address.

```

use_default_column_mapping : boolean, optional

Defaults to true, where the existing column mapping on the input table will be used. If false, a custom column mapping must be provided.

perform_ncoa : boolean, optional

Whether to update addresses for records matching the National Change of Address (NCOA) database.

ncoa_credential_id : integer, optional

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

output_level : string, optional

The set of fields persisted by a CASS or NCOA enhancement. For CASS enhancements, one of 'cass' or 'all.' For NCOA enhancements, one of 'cass', 'ncoa', 'coalesced' or 'all'. By default, all fields will be returned.

limiting_sql : string, optional

The limiting sql.

Returns id : integer

The ID for the enhancement.

name : string

The name of the enhancement.

type : string

The type of the enhancement (e.g CASS-NCOA)

created_at : string/time

The time this enhancement was created.

updated_at : string/time

The time the enhancement was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the enhancement's last run

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳hour
```

parent_id : integer

Parent ID that triggers this enhancement.

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

source : dict:

```
- database_table : dict::
    - schema : string
        The schema name of the source table.
    - table : string
        The name of the source table.
    - remote_host_id : integer
        The ID of the database host for the table.
    - credential_id : integer
        The id of the credentials to be used when performing the
        enhancement.
    - multipart_key : list
        The source table primary key.
```

destination : dict:

```
- database_table : dict::
    - schema : string
        The schema name for the output data.
    - table : string
        The table name for the output data.
```

column_mapping : dict:

```
- address1 : string
    The first address line.
- address2 : string
    The second address line.
- city : string
    The city of an address.
- state : string
    The state of an address.
- zip : string
    The zip code of an address.
- name : string
    The full name of the resident at this address. If needed, ↪
    separate
    multiple columns with `+`, e.g. `first_name+last_name`
- company : string
    The name of the company located at this address.
```

use_default_column_mapping : boolean

Defaults to true, where the existing column mapping on the input table will be used. If false, a custom column mapping must be provided.

perform_ncoa : boolean

Whether to update addresses for records matching the National Change of Address (NCOA) database.

ncoa_credential_id : integer

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

output_level : string

The set of fields persisted by a CASS or NCOA enhancement. For CASS enhancements, one of 'cass' or 'all'. For NCOA enhancements, one of 'cass', 'ncoa', 'coalesced' or 'all'. By default, all fields will be returned.

limiting_sql : string

The limiting sql.

archived : string

The archival status of the requested object(s).

```
post_cass_ncoa(name, source, *, schedule='DEFAULT', parent_id='DEFAULT', notifications='DEFAULT', destination='DEFAULT', column_mapping='DEFAULT', use_default_column_mapping='DEFAULT', perform_ncoa='DEFAULT', ncoa_credential_id='DEFAULT', output_level='DEFAULT', limiting_sql='DEFAULT')
```

Create a CASS/NCOA Enhancement

Parameters name : string

The name of the enhancement.

source : dict:

```
- database_table : dict::
  - schema : string
    The schema name of the source table.
  - table : string
    The name of the source table.
  - remote_host_id : integer
    The ID of the database host for the table.
  - credential_id : integer
    The id of the credentials to be used when performing the enhancement.
  - multipart_key : list
    The source table primary key.
```

schedule : dict, optional:

```
- scheduled : boolean
  If the object is scheduled
- scheduled_days : list
  Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
  Hours of the day it is scheduled on
- scheduled_minutes : list
  Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
  Alternative to scheduled minutes, number of times to run per hour
```

parent_id : integer, optional

Parent ID that triggers this enhancement.

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes.
↳successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author.
↳of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

destination : dict, optional:

```
- database_table : dict::
    - schema : string
        The schema name for the output data.
    - table : string
        The table name for the output data.
```

column_mapping : dict, optional:

```
- address1 : string
    The first address line.
- address2 : string
    The second address line.
- city : string
    The city of an address.
- state : string
    The state of an address.
- zip : string
    The zip code of an address.
- name : string
    The full name of the resident at this address. If needed,
↳separate
    multiple columns with `+`, e.g. `first_name+last_name`
- company : string
    The name of the company located at this address.
```

use_default_column_mapping : boolean, optional

Defaults to true, where the existing column mapping on the input table will be used. If false, a custom column mapping must be provided.

perform_ncoa : boolean, optional

Whether to update addresses for records matching the National Change of Address (NCOA) database.

ncoa_credential_id : integer, optional

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

output_level : string, optional

The set of fields persisted by a CASS or NCOA enhancement. For CASS enhancements, one of 'cass' or 'all.' For NCOA enhancements, one of 'cass', 'ncoa', 'coalesced' or 'all'. By default, all fields will be returned.

limiting_sql : string, optional

The limiting sql.

Returns id : integer

The ID for the enhancement.

name : string

The name of the enhancement.

type : string

The type of the enhancement (e.g CASS-NCOA)

created_at : string/time

The time this enhancement was created.

updated_at : string/time

The time the enhancement was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the enhancement's last run

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

parent_id : integer

Parent ID that triggers this enhancement.

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

source : dict:

```
- database_table : dict::
    - schema : string
        The schema name of the source table.
    - table : string
        The name of the source table.
- remote_host_id : integer
    The ID of the database host for the table.
- credential_id : integer
    The id of the credentials to be used when performing the
    enhancement.
```

```
- multipart_key : list
    The source table primary key.
```

destination : dict:

```
- database_table : dict::
    - schema : string
      The schema name for the output data.
    - table : string
      The table name for the output data.
```

column_mapping : dict:

```
- address1 : string
    The first address line.
- address2 : string
    The second address line.
- city : string
    The city of an address.
- state : string
    The state of an address.
- zip : string
    The zip code of an address.
- name : string
    The full name of the resident at this address. If needed, ↵
↵separate
    multiple columns with `+`, e.g. `first_name+last_name`
- company : string
    The name of the company located at this address.
```

use_default_column_mapping : boolean

Defaults to true, where the existing column mapping on the input table will be used. If false, a custom column mapping must be provided.

perform_ncoa : boolean

Whether to update addresses for records matching the National Change of Address (NCOA) database.

ncoa_credential_id : integer

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

output_level : string

The set of fields persisted by a CASS or NCOA enhancement. For CASS enhancements, one of 'cass' or 'all.' For NCOA enhancements, one of 'cass', 'ncoa', 'coalesced' or 'all'. By default, all fields will be returned.

limiting_sql : string

The limiting sql.

archived : string

The archival status of the requested object(s).

post_cass_ncoa_cancel (*id*)

Cancel a run

Parameters id : integer

The ID of the job.

Returns **id** : integer

The ID of the run.

state : string

The state of the run, one of 'queued', 'running' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

post_cass_ncoa_runs (*id*)

Start a run

Parameters **id** : integer

The ID of the cass_ncoa.

Returns **id** : integer

The ID of the run.

cass_ncoa_id : integer

The ID of the cass_ncoa.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

put_cass_ncoa (*id, name, source, *, schedule='DEFAULT', parent_id='DEFAULT', notifications='DEFAULT', destination='DEFAULT', column_mapping='DEFAULT', use_default_column_mapping='DEFAULT', perform_ncoa='DEFAULT', ncoa_credential_id='DEFAULT', output_level='DEFAULT', limiting_sql='DEFAULT'*)

Replace all attributes of this CASS/NCOA Enhancement

Parameters **id** : integer

The ID for the enhancement.

name : string

The name of the enhancement.

source : dict:

```
- database_table : dict::
  - schema : string
    The schema name of the source table.
  - table : string
    The name of the source table.
  - remote_host_id : integer
    The ID of the database host for the table.
  - credential_id : integer
    The id of the credentials to be used when performing the
    enhancement.
  - multipart_key : list
    The source table primary key.
```

schedule : dict, optional:

```
- scheduled : boolean
  If the object is scheduled
- scheduled_days : list
  Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
  Hours of the day it is scheduled on
- scheduled_minutes : list
  Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
  Alternative to scheduled minutes, number of times to run per_
↳hour
```

parent_id : integer, optional

Parent ID that triggers this enhancement.

notifications : dict, optional:

```
- urls : list
  URLs to receive a POST request at job completion
- success_email_subject : string
  Custom subject line for success e-mail.
- success_email_body : string
  Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
  Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_from_name : string
  Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
  Address for replies to success emails; defaults to the author_
↳of the
  job.
- failure_email_addresses : list
  Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
  Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
  If success email notifications are on.
- failure_on : boolean
  If failure email notifications are on.
```

destination : dict, optional:

```
- database_table : dict::
  - schema : string
    The schema name for the output data.
  - table : string
    The table name for the output data.
```

column_mapping : dict, optional:

```
- address1 : string
  The first address line.
- address2 : string
  The second address line.
- city : string
  The city of an address.
- state : string
  The state of an address.
- zip : string
  The zip code of an address.
- name : string
  The full name of the resident at this address. If needed, ↵
  ↵separate
  multiple columns with `+`, e.g. `first_name+last_name`
- company : string
  The name of the company located at this address.
```

use_default_column_mapping : boolean, optional

Defaults to true, where the existing column mapping on the input table will be used. If false, a custom column mapping must be provided.

perform_ncoa : boolean, optional

Whether to update addresses for records matching the National Change of Address (NCOA) database.

ncoa_credential_id : integer, optional

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

output_level : string, optional

The set of fields persisted by a CASS or NCOA enhancement. For CASS enhancements, one of 'cass' or 'all.' For NCOA enhancements, one of 'cass', 'ncoa', 'coalesced' or 'all'. By default, all fields will be returned.

limiting_sql : string, optional

The limiting sql.

Returns id : integer

The ID for the enhancement.

name : string

The name of the enhancement.

type : string

The type of the enhancement (e.g CASS-NCOA)

created_at : string/time

The time this enhancement was created.

updated_at : string/time

The time the enhancement was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the enhancement's last run

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
    hour
```

parent_id : integer

Parent ID that triggers this enhancement.

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes
    successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author
    of the job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
```

```

    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

running_as : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

source : dict:

```

- database_table : dict::
    - schema : string
        The schema name of the source table.
    - table : string
        The name of the source table.
- remote_host_id : integer
    The ID of the database host for the table.
- credential_id : integer
    The id of the credentials to be used when performing the
    enhancement.
- multipart_key : list
    The source table primary key.

```

destination : dict:

```

- database_table : dict::
    - schema : string
        The schema name for the output data.
    - table : string
        The table name for the output data.

```

column_mapping : dict:

```

- address1 : string
    The first address line.
- address2 : string
    The second address line.
- city : string
    The city of an address.
- state : string
    The state of an address.
- zip : string
    The zip code of an address.
- name : string
    The full name of the resident at this address. If needed, ↵
    ↵separate
    multiple columns with `+`, e.g. `first_name+last_name`
- company : string
    The name of the company located at this address.

```

use_default_column_mapping : boolean

Defaults to true, where the existing column mapping on the input table will be used. If false, a custom column mapping must be provided.

perform_ncoa : boolean

Whether to update addresses for records matching the National Change of Address (NCOA) database.

ncoa_credential_id : integer

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

output_level : string

The set of fields persisted by a CASS or NCOA enhancement. For CASS enhancements, one of 'cass' or 'all'. For NCOA enhancements, one of 'cass', 'ncoa', 'coalesced' or 'all'. By default, all fields will be returned.

limiting_sql : string

The limiting sql.

archived : string

The archival status of the requested object(s).

put_cass_ncoa_archive (*id, status*)
Update the archive status of this object

Parameters id : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns id : integer

The ID for the enhancement.

name : string

The name of the enhancement.

type : string

The type of the enhancement (e.g CASS-NCOA)

created_at : string/time

The time this enhancement was created.

updated_at : string/time

The time the enhancement was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
```

```

    This user's initials.
- online : boolean
    Whether this user is online.

```

state : string

The status of the enhancement's last run

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour

```

parent_id : integer

Parent ID that triggers this enhancement.

notifications : dict:

```

- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

running_as : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string

```

```
    This user's username.
-   initials : string
    This user's initials.
-   online : boolean
    Whether this user is online.
```

source : dict:

```
-   database_table : dict::
    -   schema : string
        The schema name of the source table.
    -   table : string
        The name of the source table.
    -   remote_host_id : integer
        The ID of the database host for the table.
    -   credential_id : integer
        The id of the credentials to be used when performing the
        enhancement.
    -   multipart_key : list
        The source table primary key.
```

destination : dict:

```
-   database_table : dict::
    -   schema : string
        The schema name for the output data.
    -   table : string
        The table name for the output data.
```

column_mapping : dict:

```
-   address1 : string
    The first address line.
-   address2 : string
    The second address line.
-   city : string
    The city of an address.
-   state : string
    The state of an address.
-   zip : string
    The zip code of an address.
-   name : string
    The full name of the resident at this address. If needed, ↵
    ↵separate
    multiple columns with `+`, e.g. `first_name+last_name`
-   company : string
    The name of the company located at this address.
```

use_default_column_mapping : boolean

Defaults to true, where the existing column mapping on the input table will be used. If false, a custom column mapping must be provided.

perform_ncoa : boolean

Whether to update addresses for records matching the National Change of Address (NCOA) database.

ncoa_credential_id : integer

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

output_level : string

The set of fields persisted by a CASS or NCOA enhancement. For CASS enhancements, one of 'cass' or 'all.' For NCOA enhancements, one of 'cass', 'ncoa', 'coalesced' or 'all'. By default, all fields will be returned.

limiting_sql : string

The limiting sql.

archived : string

The archival status of the requested object(s).

put_cass_ncoa_projects (*id, project_id*)

Add a JobTypes::CassNcoa to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_cass_ncoa_shares_groups (*id, group_ids, permission_level, *, share_email_body='DEFAULT', send_shared_email='DEFAULT'*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: "read", "write", or "manage"

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_cass_ncoa_shares_users (*id*, *user_ids*, *permission_level*,
*, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Exports

class Exports (*session_kwargs*, *return_type*='civis')

Methods

<code>list(*[, type, author, status, hidden, ...])</code>	List
---	------

list (*, *type*='DEFAULT', *author*='DEFAULT', *status*='DEFAULT', *hidden*='DEFAULT', *archived*='DEFAULT', *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List

Parameters *type* : string, optional

If specified, return exports of these types. It accepts a comma-separated list, possible values are 'database' and 'gdoc'.

author : string, optional

If specified, return exports from this author. It accepts a comma-separated list of author ids.

status : string, optional

If specified, returns export with one of these statuses. It accepts a comma-separated list, possible values are 'running', 'failed', 'succeeded', 'idle', 'scheduled'.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

archived : string, optional

The archival status of the requested object(s).

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to `updated_at`. Must be one of: `updated_at`, `name`, `created_at`, `last_run.updated_at`.

order_dir : string, optional

Direction in which to sort, either `asc` (ascending) or `desc` (descending) defaulting to `desc`.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by `limit` are needed. When True, `limit` and `page_num` are ignored. Defaults to False.

Returns **id** : integer

The ID for this export.

name : string

The name of this export.

type : string

The type of export.

created_at : string/time

The creation time for this export.

updated_at : string/time

The last modification time for this export.

state : string

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

author : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

Files

class Files (*session_kwargs*, *return_type*='civis')

Methods

<code>delete_projects(id, project_id)</code>	Remove a Data::S3File from a project
<code>delete_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>get(id, *[, link_expires_at])</code>	Get details about a file
<code>list_projects(id, *[, hidden])</code>	List the projects a Data::S3File belongs to
<code>list_shares(id)</code>	List users and groups permissioned on this object
<code>post(name, *[, expires_at])</code>	Initiate an upload of a file into the platform
<code>post_multipart(name, num_parts, *[, expires_at])</code>	Initiate a multipart upload
<code>post_multipart_complete(id)</code>	Complete a multipart upload
<code>put_projects(id, project_id)</code>	Add a Data::S3File to a project
<code>put_shares_groups(id, group_ids, ...[, ...])</code>	Set the permissions groups has on this object
<code>put_shares_users(id, user_ids, ...[, ...])</code>	Set the permissions users have on this object

delete_projects (*id*, *project_id*)
Remove a [Data::S3File](#) from a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_shares_groups (*id*, *group_id*)
Revoke the permissions a group has on this object

Parameters *id* : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users (*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters *id* : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get (*id*, *, *link_expires_at*='DEFAULT')

Get details about a file

Parameters *id* : integer

The ID of the file object.

link_expires_at : string, optional

The date and time the download link will expire. Must be a time between now and 36 hours from now. Defaults to 30 minutes from now.

Returns *id* : integer

The ID of the file object.

name : string

The file name.

created_at : string/date-time

The date and time the file was created.

file_size : integer

The file size.

expires_at : string/date-time

The date and time the file will expire. If not specified, the file will expire in 30 days. To keep a file indefinitely, specify null.

download_url : string

A JSON string containing information about the URL of the file.

file_url : string

The URL that may be used to download the file.

list_projects (*id*, *, *hidden*='DEFAULT')

List the projects a [Data::S3File](#) belongs to

Parameters *id* : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns `id` : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_shares (*id*)

List users and groups permissioned on this object

Parameters `id` : integer

The ID of the object.

Returns `readers` : dict:

```
- users : list::
    - id : integer
    - name : string
- groups : list::
    - id : integer
    - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

post (*name*, *, *expires_at*='DEFAULT')

Initiate an upload of a file into the platform

Parameters **name** : string

The file name.

expires_at : string/date-time, optional

The date and time the file will expire. If not specified, the file will expire in 30 days. To keep a file indefinitely, specify null.

Returns **id** : integer

The ID of the file object.

name : string

The file name.

created_at : string/date-time

The date and time the file was created.

file_size : integer

The file size.

expires_at : string/date-time

The date and time the file will expire. If not specified, the file will expire in 30 days. To keep a file indefinitely, specify null.

upload_url : string

The URL that may be used to upload a file. To use the upload URL, initiate a POST request to the given URL with the file you wish to import as the “file” form field.

upload_fields : dict

A hash containing the form fields to be included with the POST request.

post_multipart (*name*, *num_parts*, *, *expires_at*=*'DEFAULT'*)

Initiate a multipart upload

Parameters **name** : string

The file name.

num_parts : integer

The number of parts in which the file will be uploaded. This parameter determines the number of presigned URLs that are returned.

expires_at : string/date-time, optional

The date and time the file will expire. If not specified, the file will expire in 30 days. To keep a file indefinitely, specify null.

Returns **id** : integer

The ID of the file object.

name : string

The file name.

created_at : string/date-time

The date and time the file was created.

file_size : integer

The file size.

expires_at : string/date-time

The date and time the file will expire. If not specified, the file will expire in 30 days. To keep a file indefinitely, specify null.

upload_urls : list

An array of URLs that may be used to upload file parts. Use separate PUT requests to complete the part uploads. Links expire after 12 hours.

post_multipart_complete (*id*)

Complete a multipart upload

Parameters **id** : integer

The ID of the file object.

Returns None

Response code 204: success

put_projects (*id*, *project_id*)

Add a [Data::S3File](#) to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_shares_groups (*id*, *group_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
 send_shared_email='DEFAULT')

Set the permissions groups has on this object

Parameters *id* : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns *readers* : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT', *send_shared_email*='DEFAULT')

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns *readers* : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Groups

class Groups (*session_kwargs*, *return_type*='civis')

Methods

<code>list(*[, query, permission, limit, ...])</code>	List Groups
<hr/>	
<p>list (*, <i>query</i>='DEFAULT', <i>permission</i>='DEFAULT', <i>limit</i>='DEFAULT', <i>page_num</i>='DEFAULT', <i>order</i>='DEFAULT', <i>order_dir</i>='DEFAULT', <i>iterator</i>='DEFAULT')</p> <p>List Groups</p>	
<p>Parameters <i>query</i> : string, optional</p> <p>If specified, it will filter the groups returned. Prefix matching is supported (e.g., “query=group” will return “group” and “group of people”, but not “my group”).</p> <p>permission : string, optional</p> <p>A permissions string, one of “read”, “write”, or “manage”. Lists only groups for which the current user has that permission.</p> <p>limit : integer, optional</p> <p>Number of results to return. Defaults to 50. Maximum allowed is 1000.</p> <p>page_num : integer, optional</p> <p>Page number of the results to return. Defaults to the first page, 1.</p> <p>order : string, optional</p> <p>The field on which to order the result set. Defaults to name. Must be one of: name, created_at.</p> <p>order_dir : string, optional</p> <p>Direction in which to sort, either asc (ascending) or desc (descending) defaulting to asc.</p> <p>iterator : bool, optional</p> <p>If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.</p> <p>Returns <i>id</i> : integer</p> <p>The ID of this group.</p> <p>name : string</p> <p>This group’s name.</p> <p>created_at : string/time</p> <p>The date and time when this group was created.</p> <p>slug : string</p> <p>The slug for this group.</p> <p>organization_id : integer</p> <p>The organization associated with this group.</p>	

Imports

class Imports (*session_kwargs*, *return_type*='civis')

Methods

<code>delete_files_runs(id, run_id)</code>	Cancel a run
<code>delete_projects(id, project_id)</code>	Remove a JobTypes::Import from a project
<code>delete_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>get(id)</code>	Get details about an import
<code>get_batches(id)</code>	Get details about a batch import
<code>get_files_runs(id, run_id)</code>	Check status of a run
<code>list(*[, type, author, destination, status, ...])</code>	List
<code>list_batches(*[, hidden, limit, page_num, ...])</code>	List batch imports
<code>list_files_runs(id, *[, limit, page_num, ...])</code>	List runs for the given import
<code>list_files_runs_logs(id, run_id, *[, ...])</code>	Get the logs for a run
<code>list_projects(id, *[, hidden])</code>	List the projects a JobTypes::Import belongs to
<code>list_runs(id)</code>	Get the run history of this import
<code>list_runs_logs(id, run_id, *[, last_id, limit])</code>	Get the logs for a run
<code>list_shares(id)</code>	List users and groups permissioned on this object
<code>post(name, sync_type, is_outbound, *[, ...])</code>	Create a new import configuration
<code>post_batches(file_ids, schema, table, ...[, ...])</code>	Upload multiple files to Redshift
<code>post_cancel(id)</code>	Cancel a run
<code>post_files(schema, name, remote_host_id, ...)</code>	Initiate an import of a tabular file into the platform
<code>post_files_runs(id)</code>	Start a run
<code>post_runs(id)</code>	Run an import
<code>post_syncs(id, source, destination, *[, ...])</code>	Create a sync
<code>put(id, name, sync_type, is_outbound, *[, ...])</code>	Update an import
<code>put_archive(id, status)</code>	Update the archive status of this object
<code>put_projects(id, project_id)</code>	Add a JobTypes::Import to a project
<code>put_shares_groups(id, group_ids, ...[, ...])</code>	Set the permissions groups has on this object
<code>put_shares_users(id, user_ids, ...[, ...])</code>	Set the permissions users have on this object
<code>put_syncs(id, sync_id, source, destination, *)</code>	Update a sync
<code>put_syncs_archive(id, sync_id, *[, status])</code>	Update the archive status of this sync

delete_files_runs (*id*, *run_id*)

Cancel a run

Parameters *id* : integer

The ID of the import.

run_id : integer

The ID of the run.

Returns None

Response code 202: success

delete_projects (*id*, *project_id*)

Remove a JobTypes::Import from a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_shares_groups (*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users (*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get (*id*)

Get details about an import

Parameters **id** : integer

The ID for the import.

Returns **name** : string

The name of the import.

sync_type : string

The type of sync to perform; one of DbSync, AutoImport, SilverpopDataImport, SilverpopContactImport, GdocImport, GdocExport, and Salesforce.

source : dict:

```
- remote_host_id : integer
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific
    imports. For
    salesforce imports, the first and only element is the client
    credential
    id.
- name : string
```

destination : dict:

```
- remote_host_id : integer
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific
↳ imports. For
    salesforce imports, the first and only element is the client
↳ credential
    id.
- name : string
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
↳ hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes
↳ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author
↳ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

parent_id : integer

Parent id to trigger this import from

id : integer

The ID for the import.

is_outbound : boolean

job_type : string

The job type of this import.

syncs : list:

```
List of syncs.
- id : integer
- source : dict::
  - id : integer
    The ID of the table or file, if available.
  - path : string
    The path of the dataset to sync from; for a database source,
    schema.tablename. If you are doing a Google Sheet export,
↪this can be blank. This is a legacy parameter, it is recommended you
↪use one of the following: databaseTable, file, googleWorksheet,
↪salesforce, silverpop
  - database_table : dict::
    - schema : string
      The database schema name.
    - table : string
      The database table name.
  - file : dict::
    - id : integer
      The file id.
  - google_worksheet : dict::
    - spreadsheet : string
      The spreadsheet document name.
    - worksheet : string
      The worksheet tab name.
  - salesforce : dict::
    - object_name : string
      The Salesforce object name.
  - silverpop : dict::
    - list_id : integer
      The Silverpop list id.
- destination : dict::
  - path : string
    The schema.tablename to sync to. If you are doing a Google
↪Sheet export, this is the spreadsheet and sheet name separated by
↪a period. i.e. if you have a spreadsheet named "MySpreadsheet
↪" and a sheet called "Sheet1" this field would be "MySpreadsheet.
↪Sheet1".
    This is a legacy parameter, it is recommended you use one
↪of the following: databaseTable, googleWorksheet
  - database_table : dict::
    - schema : string
      The database schema name.
    - table : string
      The database table name.
  - google_worksheet : dict::
```



```

- spreadsheet : string
    The spreadsheet document name.
- worksheet : string
    The worksheet tab name.
- advanced_options : dict::
    - max_errors : integer
    - existing_table_rows : string
    - distkey : string
    - sortkey1 : string
    - sortkey2 : string
    - column_delimiter : string
    - column_overrides : dict
        Hash used for overriding auto-detected names and types,
        ↪with keys
            being the current name of the column being overridden.
    - identity_column : string
    - row_chunk_size : integer
    - wipe_destination_table : boolean
    - truncate_long_lines : boolean
    - invalid_char_replacement : string
    - verify_table_row_counts : boolean
    - partition_column_name : string
    - partition_schema_name : string
    - partition_table_name : string
    - partition_table_partition_column_min_name : string
    - partition_table_partition_column_max_name : string
    - last_modified_column : string
    - mysql_catalog_matches_schema : boolean
    - chunking_method : string
        The method used to break the data into smaller chunks for
        ↪transfer.
        The value can be set to sorted_by_identity_columns or if
        ↪not set
            the chunking method will be choosen automatically.
    - first_row_is_header : boolean
    - export_action : string
        The kind of export action you want to have the export
        ↪execute. Set
            to "newsprsh" if you want a new worksheet inside a new
            spreadsheet. Set to "newwksht" if you want a new worksheet
            ↪inside
                an existing spreadsheet. Set to "updatewksht" if you want to
                overwrite an existing worksheet inside an existing
                ↪spreadsheet. Set
                    to "appendwksht" if you want to append to the end of an
                    ↪existing
                        worksheet inside an existing spreadsheet.
    - sql_query : string
        If you are doing a Google Sheet export, this is your SQL
        ↪query.
    - contact_lists : string
    - soql_query : string

```

state : string

created_at : string/date-time

updated_at : string/date-time

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

user : dict:

```
- id : integer
  The ID of this user.
- name : string
  This user's name.
- username : string
  This user's username.
- initials : string
  This user's initials.
- online : boolean
  Whether this user is online.
```

running_as : dict:

```
- id : integer
  The ID of this user.
- name : string
  This user's name.
- username : string
  This user's username.
- initials : string
  This user's initials.
- online : boolean
  Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this import.

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

get_batches (*id*)

Get details about a batch import

Parameters *id* : integer

The ID for the import.

Returns *id* : integer

The ID for the import.

schema : string

The destination schema name. This schema must already exist in Redshift.

table : string

The destination table name, without the schema prefix. This table must already exist in Redshift.

remote_host_id : integer

The ID of the destination database host.

state : string

The state of the run; one of “queued”, “running”, “succeeded”, “failed”, or “cancelled”.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error returned by the run, if any.

hidden : boolean

The hidden status of the object.

get_files_runs (*id*, *run_id*)

Check status of a run

Parameters **id** : integer

The ID of the import.

run_id : integer

The ID of the run.

Returns **id** : integer

The ID of the run.

import_id : integer

The ID of the import.

state : string

The state of the run, one of ‘queued’ ‘running’ ‘succeeded’ ‘failed’ or ‘cancelled’.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

```
list (*, type='DEFAULT', author='DEFAULT', destination='DEFAULT', status='DEFAULT',  
      hidden='DEFAULT', archived='DEFAULT', limit='DEFAULT', page_num='DEFAULT',  
      order='DEFAULT', order_dir='DEFAULT', iterator='DEFAULT')
```

List

Parameters **type** : string, optional

If specified, return imports of these types. It accepts a comma-separated list, possible values are 'AutoImport', 'DbSync', 'Salesforce', 'GdocImport'.

author : string, optional

If specified, return imports from this author. It accepts a comma-separated list of author ids.

destination : string, optional

If specified, returns imports with one of these destinations. It accepts a comma-separated list of remote host ids.

status : string, optional

If specified, returns imports with one of these statuses. It accepts a comma-separated list, possible values are 'running', 'failed', 'succeeded', 'idle', 'scheduled'.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

archived : string, optional

The archival status of the requested object(s).

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to updated_at. Must be one of: updated_at, name, created_at, last_run.updated_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **name** : string

The name of the import.

sync_type : string

The type of sync to perform; one of DbSync, AutoImport, SilverpopDataImport, SilverpopContactImport, GdocImport, GdocExport, and Salesforce.

source : dict:

```
- remote_host_id : integer
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
↳ imports. For
    salesforce imports, the first and only element is the client_
↳ credential
    id.
- name : string
```

destination : dict:

```
- remote_host_id : integer
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
↳ imports. For
    salesforce imports, the first and only element is the client_
↳ credential
    id.
- name : string
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳ hour
```

id : integer

The ID for the import.

is_outbound : boolean

job_type : string

The job type of this import.

state : string

created_at : string/date-time

updated_at : string/date-time

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
```

```
The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

user : dict:

```
- id : integer
  The ID of this user.
- name : string
  This user's name.
- username : string
  This user's username.
- initials : string
  This user's initials.
- online : boolean
  Whether this user is online.
```

time_zone : string

The time zone of this import.

archived : string

The archival status of the requested object(s).

list_batches (*, *hidden='DEFAULT', limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', iterator='DEFAULT'*)

List batch imports

Parameters **hidden** : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to updated_at. Must be one of: updated_at, created_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The ID for the import.

schema : string

The destination schema name. This schema must already exist in Redshift.

table : string

The destination table name, without the schema prefix. This table must already exist in Redshift.

remote_host_id : integer

The ID of the destination database host.

state : string

The state of the run; one of “queued”, “running”, “succeeded”, “failed”, or “cancelled”.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error returned by the run, if any.

list_files_runs (*id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List runs for the given import

Parameters id : integer

The ID of the import.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 100.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns id : integer

The ID of the run.

import_id : integer

The ID of the import.

state : string

The state of the run, one of ‘queued’ ‘running’ ‘succeeded’ ‘failed’ or ‘cancelled’.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

list_files_runs_logs (*id, run_id, *, last_id='DEFAULT', limit='DEFAULT'*)

Get the logs for a run

Parameters id : integer

The ID of the import.

run_id : integer

The ID of the run.

last_id : integer, optional

The ID of the last log message received. Log entries with this ID value or lower will be omitted. Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.

limit : integer, optional

The maximum number of log messages to return. Default of 10000.

Returns id : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

message : string

The log message.

level : string

The level of the log. One of unknown,fatal,error,warn,info,debug.

list_projects (*id, *, hidden='DEFAULT'*)

List the projects a JobTypes::Import belongs to

Parameters id : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns id : integer

The ID for this project.

author : dict:


```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

name : string

The name of this project.

description : string

A description of the project

users : list:

```

Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_runs (*id*)

Get the run history of this import

Parameters *id* : integer

Returns *id* : integer

state : string

created_at : string/time

The time that the run was queued.

started_at : string/time

The time that the run started.

finished_at : string/time

The time that the run completed.

error : string

The error message for this run, if present.

list_runs_logs (*id*, *run_id*, *, *last_id*=*'DEFAULT'*, *limit*=*'DEFAULT'*)

Get the logs for a run

Parameters *id* : integer

The ID of the import.

run_id : integer

The ID of the run.

last_id : integer, optional

The ID of the last log message received. Log entries with this ID value or lower will be omitted. Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.

limit : integer, optional

The maximum number of log messages to return. Default of 10000.

Returns *id* : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

message : string

The log message.

level : string

The level of the log. One of unknown,fatal,error,warn,info,debug.

list_shares (*id*)

List users and groups permissioned on this object

Parameters *id* : integer

The ID of the object.

Returns *readers* : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

post (*name*, *sync_type*, *is_outbound*, *, *source*='DEFAULT', *destination*='DEFAULT', *schedule*='DEFAULT', *notifications*='DEFAULT', *parent_id*='DEFAULT', *next_run_at*='DEFAULT', *time_zone*='DEFAULT', *hidden*='DEFAULT')

Create a new import configuration

Parameters **name** : string

The name of the import.

sync_type : string

The type of sync to perform; one of DbSync, AutoImport, SilverpopDataImport, SilverpopContactImport, GdocImport, GdocExport, and Salesforce.

is_outbound : boolean

source : dict, optional:

```
- remote_host_id : integer
- credential_id : integer
- additional_credentials : list
  Array that holds additional credentials used for specific_
↳ imports. For
  salesforce imports, the first and only element is the client_
↳ credential
  id.
```

destination : dict, optional:

```
- remote_host_id : integer
- credential_id : integer
- additional_credentials : list
  Array that holds additional credentials used for specific_
↳ imports. For
  salesforce imports, the first and only element is the client_
↳ credential
  id.
```

schedule : dict, optional:

```
- scheduled : boolean
  If the object is scheduled
- scheduled_days : list
  Day based on numeric value starting at 0 for Sunday
```

```
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳hour
```

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

parent_id : integer, optional

Parent id to trigger this import from

next_run_at : string/time, optional

The time of the next scheduled run.

time_zone : string, optional

The time zone of this import.

hidden : boolean, optional

The hidden status of the object.

Returns name : string

The name of the import.

sync_type : string

The type of sync to perform; one of DbSync, AutoImport, SilverpopDataImport, SilverpopContactImport, GdocImport, GdocExport, and Salesforce.

source : dict:

```
- remote_host_id : integer
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
↳ imports. For
    salesforce imports, the first and only element is the client_
↳ credential
    id.
- name : string
```

destination : dict:

```
- remote_host_id : integer
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
↳ imports. For
    salesforce imports, the first and only element is the client_
↳ credential
    id.
- name : string
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳ hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
```

```
If success email notifications are on.
- failure_on : boolean
  If failure email notifications are on.
```

parent_id : integer

Parent id to trigger this import from

id : integer

The ID for the import.

is_outbound : boolean

job_type : string

The job type of this import.

syncs : list:

```
List of syncs.
- id : integer
- source : dict::
  - id : integer
    The ID of the table or file, if available.
  - path : string
    The path of the dataset to sync from; for a database source,
    schema.tablename. If you are doing a Google Sheet export,
↪this can
    be blank. This is a legacy parameter, it is recommended you
↪use one
    of the following: databaseTable, file, googleWorksheet,
↪salesforce,
    silverpop
  - database_table : dict::
    - schema : string
      The database schema name.
    - table : string
      The database table name.
  - file : dict::
    - id : integer
      The file id.
  - google_worksheet : dict::
    - spreadsheet : string
      The spreadsheet document name.
    - worksheet : string
      The worksheet tab name.
  - salesforce : dict::
    - object_name : string
      The Salesforce object name.
  - silverpop : dict::
    - list_id : integer
      The Silverpop list id.
- destination : dict::
  - path : string
    The schema.tablename to sync to. If you are doing a Google
↪Sheet
    export, this is the spreadsheet and sheet name separated by
↪a
    period. i.e. if you have a spreadsheet named "MySpreadsheet
↪" and a
```

```

    sheet called "Sheet1" this field would be "MySpreadsheet.
↪Sheet1".
    This is a legacy parameter, it is recommended you use one_
↪of the
        following: databaseTable, googleWorksheet
    - database_table : dict::
        - schema : string
            The database schema name.
        - table : string
            The database table name.
    - google_worksheet : dict::
        - spreadsheet : string
            The spreadsheet document name.
        - worksheet : string
            The worksheet tab name.
    - advanced_options : dict::
        - max_errors : integer
        - existing_table_rows : string
        - distkey : string
        - sortkey1 : string
        - sortkey2 : string
        - column_delimiter : string
        - column_overrides : dict
            Hash used for overriding auto-detected names and types,
↪with keys
            being the current name of the column being overridden.
        - identity_column : string
        - row_chunk_size : integer
        - wipe_destination_table : boolean
        - truncate_long_lines : boolean
        - invalid_char_replacement : string
        - verify_table_row_counts : boolean
        - partition_column_name : string
        - partition_schema_name : string
        - partition_table_name : string
        - partition_table_partition_column_min_name : string
        - partition_table_partition_column_max_name : string
        - last_modified_column : string
        - mysql_catalog_matches_schema : boolean
        - chunking_method : string
            The method used to break the data into smaller chunks for
↪transfer.
            The value can be set to sorted_by_identity_columns or if
↪not set
            the chunking method will be choosen automatically.
        - first_row_is_header : boolean
        - export_action : string
            The kind of export action you want to have the export_
↪execute. Set
            to "newsprshst" if you want a new worksheet inside a new
            spreadsheet. Set to "newwksht" if you want a new worksheet_
↪inside
            an existing spreadsheet. Set to "updatewksht" if you want to
            overwrite an existing worksheet inside an existing_
↪spreadsheet. Set
            to "appendwksht" if you want to append to the end of an_
↪existing
            worksheet inside an existing spreadsheet.

```

```
- sql_query : string
    If you are doing a Google Sheet export, this is your SQL
↪query.
- contact_lists : string
- soql_query : string
```

state : string

created_at : string/date-time

updated_at : string/date-time

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this import.

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

post_batches (*file_ids*, *schema*, *table*, *remote_host_id*, *credential_id*, *, *column_delimiter*=*'DEFAULT'*, *first_row_is_header*=*'DEFAULT'*, *compression*=*'DEFAULT'*, *hidden*=*'DEFAULT'*)

Upload multiple files to Redshift

Parameters **file_ids** : list

The file IDs for the import.

schema : string

The destination schema name. This schema must already exist in Redshift.

table : string

The destination table name, without the schema prefix. This table must already exist in Redshift.

remote_host_id : integer

The ID of the destination database host.

credential_id : integer

The ID of the credentials to be used when performing the database import.

column_delimiter : string, optional

The column delimiter for the file. Valid arguments are “comma”, “tab”, and “pipe”. If unspecified, defaults to “comma”.

first_row_is_header : boolean, optional

A boolean value indicating whether or not the first row is a header row. If unspecified, defaults to false.

compression : string, optional

The type of compression. Valid arguments are “gzip”, “zip”, and “none”. If unspecified, defaults to “gzip”.

hidden : boolean, optional

The hidden status of the object.

Returns **id** : integer

The ID for the import.

schema : string

The destination schema name. This schema must already exist in Redshift.

table : string

The destination table name, without the schema prefix. This table must already exist in Redshift.

remote_host_id : integer

The ID of the destination database host.

state : string

The state of the run; one of “queued”, “running”, “succeeded”, “failed”, or “cancelled”.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error returned by the run, if any.

hidden : boolean

The hidden status of the object.

post_cancel (*id*)

Cancel a run

Parameters id : integer

The ID of the job.

Returns id : integer

The ID of the run.

state : string

The state of the run, one of ‘queued’, ‘running’ or ‘cancelled’.

is_cancel_requested : boolean

True if run cancel requested, else false.

post_files (*schema, name, remote_host_id, credential_id, *, max_errors='DEFAULT', existing_table_rows='DEFAULT', distkey='DEFAULT', sortkey1='DEFAULT', sortkey2='DEFAULT', column_delimiter='DEFAULT', first_row_is_header='DEFAULT', multipart='DEFAULT', hidden='DEFAULT'*)

Initiate an import of a tabular file into the platform

Parameters schema : string

The schema of the destination table.

name : string

The name of the destination table.

remote_host_id : integer

The id of the destination database host.

credential_id : integer

The id of the credentials to be used when performing the database import.

max_errors : integer, optional

The maximum number of rows with errors to remove from the import before failing.

existing_table_rows : string, optional

The behaviour if a table with the requested name already exists. One of “fail”, “truncate”, “append”, or “drop”. Defaults to “fail”.

distkey : string, optional

The column to use as the distkey for the table.

sortkey1 : string, optional

The column to use as the sort key for the table.

sortkey2 : string, optional

The second column in a compound sortkey for the table.

column_delimiter : string, optional

The column delimiter of the file. If `column_delimiter` is null or omitted, it will be auto-detected. Valid arguments are “comma”, “tab”, and “pipe”.

first_row_is_header : boolean, optional

A boolean value indicating whether or not the first row is a header row. If `first_row_is_header` is null or omitted, it will be auto-detected.

multipart : boolean, optional

If true, the upload URI will require a *multipart/form-data* POST request. Defaults to false.

hidden : boolean, optional

The hidden status of the object.

Returns id : integer

The id of the import.

upload_uri : string

The URI which may be used to upload a tabular file for import. You must use this URI to upload the file you wish imported and then inform the Civis API when your upload is complete using the URI given by the `runUri` field of this response.

run_uri : string

The URI to POST to once the file upload is complete. After uploading the file using the URI given in the `uploadUri` attribute of the response, POST to this URI to initiate the import of your uploaded file into the platform.

upload_fields : dict

If `multipart` was set to true, these fields should be included in the multipart upload.

post_files_runs (*id*)

Start a run

Parameters id : integer

The ID of the import.

Returns id : integer

The ID of the run.

import_id : integer

The ID of the import.

state : string

The state of the run, one of ‘queued’ ‘running’ ‘succeeded’ ‘failed’ or ‘cancelled’.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

post_runs (*id*)

Run an import

Parameters **id** : integer

The ID of the import to run.

Returns **run_id** : integer

The ID of the new run triggered.

post_syncs (*id, source, destination, *, advanced_options='DEFAULT'*)

Create a sync

Parameters **id** : integer

source : dict:

```
- path : string
    The path of the dataset to sync from; for a database source,
    schema.tablename. If you are doing a Google Sheet export, this
    ↪can be
    blank. This is a legacy parameter, it is recommended you use
    ↪one of the
    following: databaseTable, file, googleWorksheet, salesforce,
    ↪silverpop
- database_table : dict::
    - schema : string
        The database schema name.
    - table : string
        The database table name.
- file : dict
- google_worksheet : dict::
    - spreadsheet : string
        The spreadsheet document name.
    - worksheet : string
        The worksheet tab name.
- salesforce : dict::
    - object_name : string
        The Salesforce object name.
- silverpop : dict::
    - list_id : integer
        The Silverpop list id.
```

destination : dict:

```
- path : string
    The schema.tablename to sync to. If you are doing a Google Sheet
    export, this is the spreadsheet and sheet name separated by a
    ↪period.
```

i.e. **if** you have a spreadsheet named "MySpreadsheet" and a sheet called "Sheet1" this field would be "MySpreadsheet.Sheet1". This **is** a legacy parameter, it **is** recommended you use one of the following:

- databaseTable, googleWorksheet
- database_table : dict::
 - schema : string
 - The database schema name.
 - table : string
 - The database table name.
- google_worksheet : dict::
 - spreadsheet : string
 - The spreadsheet document name.
 - worksheet : string
 - The worksheet tab name.

advanced_options : dict, optional:

- max_errors : integer
- existing_table_rows : string
- distkey : string
- sortkey1 : string
- sortkey2 : string
- column_delimiter : string
- column_overrides : dict
 - Hash used **for** overriding auto-detected names **and** types, **with** keys being
 - the current name of the column being overridden.
- identity_column : string
- row_chunk_size : integer
- wipe_destination_table : boolean
- truncate_long_lines : boolean
- invalid_char_replacement : string
- verify_table_row_counts : boolean
- partition_column_name : string
- partition_schema_name : string
- partition_table_name : string
- partition_table_partition_column_min_name : string
- partition_table_partition_column_max_name : string
- last_modified_column : string
- mysql_catalog_matches_schema : boolean
- chunking_method : string
 - The method used to **break** the data into smaller chunks **for** transfer.
 - The value can be **set** to sorted_by_identity_columns **or if not set** the chunking method will be chosen automatically.
- first_row_is_header : boolean
- export_action : string
 - The kind of export action you want to have the export execute.
 - Set to "newsprshst" **if** you want a new worksheet inside a new spreadsheet. Set to "newwksht" **if** you want a new worksheet inside an existing spreadsheet. Set to "updatewksht" **if** you want to overwrite an existing worksheet inside an existing spreadsheet. Set to "appendwksht" **if** you

```
    want to append to the end of an existing worksheet inside an
    ↪existing
      spreadsheet.
- sql_query : string
    If you are doing a Google Sheet export, this is your SQL query.
- contact_lists : string
- soql_query : string
```

Returns `id : integer`

source : dict:

```
- id : integer
    The ID of the table or file, if available.
- path : string
    The path of the dataset to sync from; for a database source,
    schema.tablename. If you are doing a Google Sheet export, this
    ↪can be
      blank. This is a legacy parameter, it is recommended you use
    ↪one of the
      following: databaseTable, file, googleWorksheet, salesforce,
    ↪silverpop
- database_table : dict::
    - schema : string
        The database schema name.
    - table : string
        The database table name.
- file : dict::
    - id : integer
        The file id.
- google_worksheet : dict::
    - spreadsheet : string
        The spreadsheet document name.
    - worksheet : string
        The worksheet tab name.
- salesforce : dict::
    - object_name : string
        The Salesforce object name.
- silverpop : dict::
    - list_id : integer
        The Silverpop list id.
```

destination : dict:

```
- path : string
    The schema.tablename to sync to. If you are doing a Google Sheet
    export, this is the spreadsheet and sheet name separated by a
    ↪period.
    i.e. if you have a spreadsheet named "MySpreadsheet" and a
    ↪sheet called
      "Sheet1" this field would be "MySpreadsheet.Sheet1". This is a
    ↪legacy
      parameter, it is recommended you use one of the following:
      databaseTable, googleWorksheet
- database_table : dict::
    - schema : string
        The database schema name.
    - table : string
```

```

    The database table name.
- google_worksheet : dict::
    - spreadsheet : string
      The spreadsheet document name.
    - worksheet : string
      The worksheet tab name.

```

advanced_options : dict:

```

- max_errors : integer
- existing_table_rows : string
- distkey : string
- sortkey1 : string
- sortkey2 : string
- column_delimiter : string
- column_overrides : dict
    Hash used for overriding auto-detected names and types, with
    ↪ keys being
      the current name of the column being overridden.
- identity_column : string
- row_chunk_size : integer
- wipe_destination_table : boolean
- truncate_long_lines : boolean
- invalid_char_replacement : string
- verify_table_row_counts : boolean
- partition_column_name : string
- partition_schema_name : string
- partition_table_name : string
- partition_table_partition_column_min_name : string
- partition_table_partition_column_max_name : string
- last_modified_column : string
- mysql_catalog_matches_schema : boolean
- chunking_method : string
    The method used to break the data into smaller chunks for
    ↪ transfer.
    The value can be set to sorted_by_identity_columns or if not
    ↪ set the
      chunking method will be choosen automatically.
- first_row_is_header : boolean
- export_action : string
    The kind of export action you want to have the export execute.
    ↪ Set to
      "newsprshst" if you want a new worksheet inside a new
    ↪ spreadsheet. Set
      to "newwksht" if you want a new worksheet inside an existing
      spreadsheet. Set to "updatewksht" if you want to overwrite an
    ↪ existing
      worksheet inside an existing spreadsheet. Set to "appendwksht"
    ↪ if you
      want to append to the end of an existing worksheet inside an
    ↪ existing
      spreadsheet.
- sql_query : string
    If you are doing a Google Sheet export, this is your SQL query.
- contact_lists : string
- soql_query : string

```

put (*id*, *name*, *sync_type*, *is_outbound*, *, *source*='DEFAULT', *destination*='DEFAULT', *schedule*='DEFAULT', *notifications*='DEFAULT', *parent_id*='DEFAULT', *next_run_at*='DEFAULT', *time_zone*='DEFAULT')

Update an import

Parameters *id* : integer

The ID for the import.

name : string

The name of the import.

sync_type : string

The type of sync to perform; one of DbSync, AutoImport, SilverpopDataImport, SilverpopContactImport, GdocImport, GdocExport, and Salesforce.

is_outbound : boolean

source : dict, optional:

```
- remote_host_id : integer
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
↳ imports. For
    salesforce imports, the first and only element is the client_
↳ credential
    id.
```

destination : dict, optional:

```
- remote_host_id : integer
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
↳ imports. For
    salesforce imports, the first and only element is the client_
↳ credential
    id.
```

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳ hour
```

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
```



```

    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
    ↪ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
    ↪ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

parent_id : integer, optional

Parent id to trigger this import from

next_run_at : string/time, optional

The time of the next scheduled run.

time_zone : string, optional

The time zone of this import.

Returns name : string

The name of the import.

sync_type : string

The type of sync to perform; one of DbSync, AutoImport, SilverpopDataImport, SilverpopContactImport, GdocImport, GdocExport, and Salesforce.

source : dict:

```

- remote_host_id : integer
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific,
    ↪ imports. For
    salesforce imports, the first and only element is the client,
    ↪ credential
    id.
- name : string

```

destination : dict:

```

- remote_host_id : integer
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific,
    ↪ imports. For

```

```
    salesforce imports, the first and only element is the client_  
↪credential  
    id.  
- name : string
```

schedule : dict:

```
- scheduled : boolean  
    If the object is scheduled  
- scheduled_days : list  
    Day based on numeric value starting at 0 for Sunday  
- scheduled_hours : list  
    Hours of the day it is scheduled on  
- scheduled_minutes : list  
    Minutes of the day it is scheduled on  
- scheduled_runs_per_hour : integer  
    Alternative to scheduled minutes, number of times to run per_  
↪hour
```

notifications : dict:

```
- urls : list  
    URLs to receive a POST request at job completion  
- success_email_subject : string  
    Custom subject line for success e-mail.  
- success_email_body : string  
    Custom body text for success e-mail, written in Markdown.  
- success_email_addresses : list  
    Addresses to notify by e-mail when the job completes_  
↪successfully.  
- success_email_from_name : string  
    Name from which success emails are sent; defaults to "Civis."  
- success_email_reply_to : string  
    Address for replies to success emails; defaults to the author_  
↪of the  
    job.  
- failure_email_addresses : list  
    Addresses to notify by e-mail when the job fails.  
- stall_warning_minutes : integer  
    Stall warning emails will be sent after this amount of minutes.  
- success_on : boolean  
    If success email notifications are on.  
- failure_on : boolean  
    If failure email notifications are on.
```

parent_id : integer

Parent id to trigger this import from

id : integer

The ID for the import.

is_outbound : boolean

job_type : string

The job type of this import.

syncs : list:

```

List of syncs.
- id : integer
- source : dict::
  - id : integer
    The ID of the table or file, if available.
  - path : string
    The path of the dataset to sync from; for a database source,
    schema.tablename. If you are doing a Google Sheet export,
↪this can
    be blank. This is a legacy parameter, it is recommended you
↪use one
    of the following: databaseTable, file, googleWorksheet,
↪salesforce,
    silverpop
  - database_table : dict::
    - schema : string
      The database schema name.
    - table : string
      The database table name.
  - file : dict::
    - id : integer
      The file id.
  - google_worksheet : dict::
    - spreadsheet : string
      The spreadsheet document name.
    - worksheet : string
      The worksheet tab name.
  - salesforce : dict::
    - object_name : string
      The Salesforce object name.
  - silverpop : dict::
    - list_id : integer
      The Silverpop list id.
- destination : dict::
  - path : string
    The schema.tablename to sync to. If you are doing a Google
↪Sheet
    export, this is the spreadsheet and sheet name separated by
↪a
    period. i.e. if you have a spreadsheet named "MySpreadsheet
↪" and a
    sheet called "Sheet1" this field would be "MySpreadsheet.
↪Sheet1".
    This is a legacy parameter, it is recommended you use one
↪of the
    following: databaseTable, googleWorksheet
  - database_table : dict::
    - schema : string
      The database schema name.
    - table : string
      The database table name.
  - google_worksheet : dict::
    - spreadsheet : string
      The spreadsheet document name.
    - worksheet : string
      The worksheet tab name.
- advanced_options : dict::
  - max_errors : integer

```

```

- existing_table_rows : string
- distkey : string
- sortkey1 : string
- sortkey2 : string
- column_delimiter : string
- column_overrides : dict
    Hash used for overriding auto-detected names and types,
↪with keys
    being the current name of the column being overridden.
- identity_column : string
- row_chunk_size : integer
- wipe_destination_table : boolean
- truncate_long_lines : boolean
- invalid_char_replacement : string
- verify_table_row_counts : boolean
- partition_column_name : string
- partition_schema_name : string
- partition_table_name : string
- partition_table_partition_column_min_name : string
- partition_table_partition_column_max_name : string
- last_modified_column : string
- mysql_catalog_matches_schema : boolean
- chunking_method : string
    The method used to break the data into smaller chunks for
↪transfer.
    The value can be set to sorted_by_identity_columns or if
↪not set
    the chunking method will be choosen automatically.
- first_row_is_header : boolean
- export_action : string
    The kind of export action you want to have the export
↪execute. Set
    to "newsprsh" if you want a new worksheet inside a new
    spreadsheet. Set to "newwksht" if you want a new worksheet
↪inside
    an existing spreadsheet. Set to "updatewksht" if you want to
    overwrite an existing worksheet inside an existing
↪spreadsheet. Set
    to "appendwksht" if you want to append to the end of an
↪existing
    worksheet inside an existing spreadsheet.
- sql_query : string
    If you are doing a Google Sheet export, this is your SQL
↪query.
- contact_lists : string
- soql_query : string

```

state : string

created_at : string/date-time

updated_at : string/date-time

last_run : dict:

```

- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.

```

```

- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

user : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

running_as : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this import.

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

put_archive (*id, status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **name** : string

The name of the import.

sync_type : string

The type of sync to perform; one of DbSync, AutoImport, SilverpopDataImport, SilverpopContactImport, GdocImport, GdocExport, and Salesforce.

source : dict:

```
- remote_host_id : integer
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
↳ imports. For
    salesforce imports, the first and only element is the client_
↳ credential
    id.
- name : string
```

destination : dict:

```
- remote_host_id : integer
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
↳ imports. For
    salesforce imports, the first and only element is the client_
↳ credential
    id.
- name : string
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳ hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳ of the
```

```

    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

parent_id : integer

Parent id to trigger this import from

id : integer

The ID for the import.

is_outbound : boolean

job_type : string

The job type of this import.

syncs : list:

```

List of syncs.
- id : integer
- source : dict::
    - id : integer
        The ID of the table or file, if available.
    - path : string
        The path of the dataset to sync from; for a database source,
        schema.tablename. If you are doing a Google Sheet export,
↪this can
        be blank. This is a legacy parameter, it is recommended you
↪use one
        of the following: databaseTable, file, googleWorksheet,
↪salesforce,
        silverpop
    - database_table : dict::
        - schema : string
            The database schema name.
        - table : string
            The database table name.
    - file : dict::
        - id : integer
            The file id.
    - google_worksheet : dict::
        - spreadsheet : string
            The spreadsheet document name.
        - worksheet : string
            The worksheet tab name.
    - salesforce : dict::
        - object_name : string
            The Salesforce object name.
    - silverpop : dict::
        - list_id : integer
            The Silverpop list id.
- destination : dict::

```

```
- path : string
    The schema.tablename to sync to. If you are doing a Google_
↳Sheet
    export, this is the spreadsheet and sheet name separated by_
↳a
    period. i.e. if you have a spreadsheet named "MySpreadsheet
↳" and a
    sheet called "Sheet1" this field would be "MySpreadsheet.
↳Sheet1".
    This is a legacy parameter, it is recommended you use one_
↳of the
    following: databaseTable, googleWorksheet
- database_table : dict::
    - schema : string
        The database schema name.
    - table : string
        The database table name.
- google_worksheet : dict::
    - spreadsheet : string
        The spreadsheet document name.
    - worksheet : string
        The worksheet tab name.
- advanced_options : dict::
    - max_errors : integer
    - existing_table_rows : string
    - distkey : string
    - sortkey1 : string
    - sortkey2 : string
    - column_delimiter : string
    - column_overrides : dict
        Hash used for overriding auto-detected names and types,
↳with keys
        being the current name of the column being overridden.
    - identity_column : string
    - row_chunk_size : integer
    - wipe_destination_table : boolean
    - truncate_long_lines : boolean
    - invalid_char_replacement : string
    - verify_table_row_counts : boolean
    - partition_column_name : string
    - partition_schema_name : string
    - partition_table_name : string
    - partition_table_partition_column_min_name : string
    - partition_table_partition_column_max_name : string
    - last_modified_column : string
    - mysql_catalog_matches_schema : boolean
    - chunking_method : string
        The method used to break the data into smaller chunks for
↳transfer.
        The value can be set to sorted_by_identity_columns or if
↳not set
        the chunking method will be choosen automatically.
    - first_row_is_header : boolean
    - export_action : string
        The kind of export action you want to have the export_
↳execute. Set
        to "newsprsh" if you want a new worksheet inside a new
        spreadsheet. Set to "newwksht" if you want a new worksheet_
↳inside
```



```

        an existing spreadsheet. Set to "updatewsht" if you want to
        overwrite an existing worksheet inside an existing
↪ spreadsheet. Set
        to "appendwsht" if you want to append to the end of an
↪ existing
        worksheet inside an existing spreadsheet.
        - sql_query : string
          If you are doing a Google Sheet export, this is your SQL
↪ query.
        - contact_lists : string
        - soql_query : string

```

state : string

created_at : string/date-time

updated_at : string/date-time

last_run : dict:

```

- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.

```

user : dict:

```

- id : integer
  The ID of this user.
- name : string
  This user's name.
- username : string
  This user's username.
- initials : string
  This user's initials.
- online : boolean
  Whether this user is online.

```

running_as : dict:

```

- id : integer
  The ID of this user.
- name : string
  This user's name.
- username : string
  This user's username.
- initials : string
  This user's initials.
- online : boolean
  Whether this user is online.

```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this import.

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

put_projects (*id*, *project_id*)

Add a JobTypes::Import to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_shares_groups (*id*, *group_ids*, *permission_level*, *, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
```

```
- id : integer
- name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT', *send_shared_email*='DEFAULT')

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_syncs (*id*, *sync_id*, *source*, *destination*, *, *advanced_options*='DEFAULT')

Update a sync

Parameters *id* : integer

The ID of the import to fetch.

sync_id : integer

The ID of the sync to fetch.

source : dict:

```
- path : string
  The path of the dataset to sync from; for a database source,
  schema.tablename. If you are doing a Google Sheet export, this
  ↪can be
  blank. This is a legacy parameter, it is recommended you use
  ↪one of the
  following: databaseTable, file, googleWorksheet, salesforce,
  ↪silverpop
- database_table : dict::
  - schema : string
    The database schema name.
  - table : string
    The database table name.
- file : dict
- google_worksheet : dict::
  - spreadsheet : string
    The spreadsheet document name.
  - worksheet : string
    The worksheet tab name.
- salesforce : dict::
  - object_name : string
    The Salesforce object name.
- silverpop : dict::
  - list_id : integer
    The Silverpop list id.
```

destination : dict:

```
- path : string
  The schema.tablename to sync to. If you are doing a Google Sheet
```

```

    export, this is the spreadsheet and sheet name separated by a
    ↪period.
    i.e. if you have a spreadsheet named "MySpreadsheet" and a
    ↪sheet called
    "Sheet1" this field would be "MySpreadsheet.Sheet1". This is a
    ↪legacy
    parameter, it is recommended you use one of the following:
    databaseTable, googleWorksheet
- database_table : dict::
  - schema : string
    The database schema name.
  - table : string
    The database table name.
- google_worksheet : dict::
  - spreadsheet : string
    The spreadsheet document name.
  - worksheet : string
    The worksheet tab name.

```

advanced_options : dict, optional:

```

- max_errors : integer
- existing_table_rows : string
- distkey : string
- sortkey1 : string
- sortkey2 : string
- column_delimiter : string
- column_overrides : dict
  Hash used for overriding auto-detected names and types, with
  ↪keys being
  the current name of the column being overridden.
- identity_column : string
- row_chunk_size : integer
- wipe_destination_table : boolean
- truncate_long_lines : boolean
- invalid_char_replacement : string
- verify_table_row_counts : boolean
- partition_column_name : string
- partition_schema_name : string
- partition_table_name : string
- partition_table_partition_column_min_name : string
- partition_table_partition_column_max_name : string
- last_modified_column : string
- mysql_catalog_matches_schema : boolean
- chunking_method : string
  The method used to break the data into smaller chunks for
  ↪transfer.
  The value can be set to sorted_by_identity_columns or if not
  ↪set the
  chunking method will be chosen automatically.
- first_row_is_header : boolean
- export_action : string
  The kind of export action you want to have the export execute.
  ↪Set to
  "newsprsh" if you want a new worksheet inside a new
  ↪spreadsheet. Set
  to "newwksht" if you want a new worksheet inside an existing
  spreadsheet. Set to "updatewksht" if you want to overwrite an
  ↪existing

```

```
    worksheet inside an existing spreadsheet. Set to "appendwksht"
↪if you
    want to append to the end of an existing worksheet inside an
↪existing
    spreadsheet.
- sql_query : string
    If you are doing a Google Sheet export, this is your SQL query.
- contact_lists : string
- soql_query : string
```

Returns `id` : integer

source : dict:

```
- id : integer
    The ID of the table or file, if available.
- path : string
    The path of the dataset to sync from; for a database source,
    schema.tablename. If you are doing a Google Sheet export, this
↪can be
    blank. This is a legacy parameter, it is recommended you use
↪one of the
    following: databaseTable, file, googleWorksheet, salesforce,
↪silverpop
- database_table : dict::
    - schema : string
        The database schema name.
    - table : string
        The database table name.
- file : dict::
    - id : integer
        The file id.
- google_worksheet : dict::
    - spreadsheet : string
        The spreadsheet document name.
    - worksheet : string
        The worksheet tab name.
- salesforce : dict::
    - object_name : string
        The Salesforce object name.
- silverpop : dict::
    - list_id : integer
        The Silverpop list id.
```

destination : dict:

```
- path : string
    The schema.tablename to sync to. If you are doing a Google Sheet
    export, this is the spreadsheet and sheet name separated by a
↪period.
    i.e. if you have a spreadsheet named "MySpreadsheet" and a
↪sheet called
    "Sheet1" this field would be "MySpreadsheet.Sheet1". This is a
↪legacy
    parameter, it is recommended you use one of the following:
    databaseTable, googleWorksheet
- database_table : dict::
    - schema : string
```

```

    The database schema name.
    - table : string
      The database table name.
- google_worksheet : dict::
    - spreadsheet : string
      The spreadsheet document name.
    - worksheet : string
      The worksheet tab name.

```

advanced_options : dict:

```

- max_errors : integer
- existing_table_rows : string
- distkey : string
- sortkey1 : string
- sortkey2 : string
- column_delimiter : string
- column_overrides : dict
    Hash used for overriding auto-detected names and types, with
    ↪ keys being
      the current name of the column being overridden.
- identity_column : string
- row_chunk_size : integer
- wipe_destination_table : boolean
- truncate_long_lines : boolean
- invalid_char_replacement : string
- verify_table_row_counts : boolean
- partition_column_name : string
- partition_schema_name : string
- partition_table_name : string
- partition_table_partition_column_min_name : string
- partition_table_partition_column_max_name : string
- last_modified_column : string
- mysql_catalog_matches_schema : boolean
- chunking_method : string
    The method used to break the data into smaller chunks for
    ↪ transfer.
    The value can be set to sorted_by_identity_columns or if not
    ↪ set the
      chunking method will be chosen automatically.
- first_row_is_header : boolean
- export_action : string
    The kind of export action you want to have the export execute.
    ↪ Set to
      "newsprsh" if you want a new worksheet inside a new
    ↪ spreadsheet. Set
      to "newwksht" if you want a new worksheet inside an existing
      spreadsheet. Set to "updatewksht" if you want to overwrite an
    ↪ existing
      worksheet inside an existing spreadsheet. Set to "appendwksht"
    ↪ if you
      want to append to the end of an existing worksheet inside an
    ↪ existing
      spreadsheet.
- sql_query : string
    If you are doing a Google Sheet export, this is your SQL query.
- contact_lists : string
- soql_query : string

```

put_syncs_archive (*id*, *sync_id*, *, *status*=*'DEFAULT'*)

Update the archive status of this sync

Parameters *id* : integer

The ID of the import to fetch.

sync_id : integer

The ID of the sync to fetch.

status : boolean, optional

The desired archived status of the sync.

Returns *id* : integer

source : dict:

```
- id : integer
    The ID of the table or file, if available.
- path : string
    The path of the dataset to sync from; for a database source,
    schema.tablename. If you are doing a Google Sheet export, this
    ↪can be
    blank. This is a legacy parameter, it is recommended you use
    ↪one of the
    following: databaseTable, file, googleWorksheet, salesforce,
    ↪silverpop
- database_table : dict::
    - schema : string
        The database schema name.
    - table : string
        The database table name.
- file : dict::
    - id : integer
        The file id.
- google_worksheet : dict::
    - spreadsheet : string
        The spreadsheet document name.
    - worksheet : string
        The worksheet tab name.
- salesforce : dict::
    - object_name : string
        The Salesforce object name.
- silverpop : dict::
    - list_id : integer
        The Silverpop list id.
```

destination : dict:

```
- path : string
    The schema.tablename to sync to. If you are doing a Google Sheet
    export, this is the spreadsheet and sheet name separated by a
    ↪period.
    i.e. if you have a spreadsheet named "MySpreadsheet" and a
    ↪sheet called
    "Sheet1" this field would be "MySpreadsheet.Sheet1". This is a
    ↪legacy
    parameter, it is recommended you use one of the following:
    databaseTable, googleWorksheet
```



```

- database_table : dict::
  - schema : string
    The database schema name.
  - table : string
    The database table name.
- google_worksheet : dict::
  - spreadsheet : string
    The spreadsheet document name.
  - worksheet : string
    The worksheet tab name.

```

advanced_options : dict:

```

- max_errors : integer
- existing_table_rows : string
- distkey : string
- sortkey1 : string
- sortkey2 : string
- column_delimiter : string
- column_overrides : dict
  Hash used for overriding auto-detected names and types, with
  ↪keys being
    the current name of the column being overridden.
- identity_column : string
- row_chunk_size : integer
- wipe_destination_table : boolean
- truncate_long_lines : boolean
- invalid_char_replacement : string
- verify_table_row_counts : boolean
- partition_column_name : string
- partition_schema_name : string
- partition_table_name : string
- partition_table_partition_column_min_name : string
- partition_table_partition_column_max_name : string
- last_modified_column : string
- mysql_catalog_matches_schema : boolean
- chunking_method : string
  The method used to break the data into smaller chunks for
  ↪transfer.
  The value can be set to sorted_by_identity_columns or if not
  ↪set the
    chunking method will be choosen automatically.
- first_row_is_header : boolean
- export_action : string
  The kind of export action you want to have the export execute.
  ↪Set to
    "newsprsh" if you want a new worksheet inside a new
  ↪spreadsheet. Set
    to "newwksht" if you want a new worksheet inside an existing
    spreadsheet. Set to "updatewksht" if you want to overwrite an
  ↪existing
    worksheet inside an existing spreadsheet. Set to "appendwksht"
  ↪if you
    want to append to the end of an existing worksheet inside an
  ↪existing
    spreadsheet.
- sql_query : string
  If you are doing a Google Sheet export, this is your SQL query.

```

```
- contact_lists : string
- soql_query : string
```

Jobs

```
class Jobs (session_kwargs, return_type='civis')
```

Methods

<code>delete_projects(id, project_id)</code>	Remove a Job from a project
<code>delete_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>get(id)</code>	Show basic job info
<code>get_runs(id, run_id)</code>	Check status of a job
<code>list(*[, state, type, q, permission, ...])</code>	List
<code>list_children(id)</code>	Show nested tree of children that this job triggers
<code>list_parents(id)</code>	Show chain of parents as a list that this job triggers from
<code>list_projects(id, *[, hidden])</code>	List the projects a Job belongs to
<code>list_shares(id)</code>	List users and groups permissioned on this object
<code>post_runs(id)</code>	Run a job
<code>post_trigger_email(id)</code>	Generate and retrieve trigger email address
<code>put_projects(id, project_id)</code>	Add a Job to a project
<code>put_shares_groups(id, group_ids, ...[, ...])</code>	Set the permissions groups has on this object
<code>put_shares_users(id, user_ids, ...[, ...])</code>	Set the permissions users have on this object

delete_projects (*id*, *project_id*)

Remove a Job from a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_shares_groups (*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users (*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters `id` : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get (*id*)

Show basic job info

Parameters `id` : integer

The ID for this job.

Returns `id` : integer

name : string

type : string

state : string

Whether the job is idle, queued, running, cancelled, or failed.

created_at : string/date-time

updated_at : string/date-time

runs : list:

Information about the most recent runs of the job.

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

get_runs (*id*, *run_id*)

Check status of a job

Parameters *id* : integer

The ID of the Job.

run_id : integer

The ID of the Run.

Returns *id* : integer

state : string

created_at : string/time

The time that the run was queued.

started_at : string/time

The time that the run started.

finished_at : string/time

The time that the run completed.

error : string

The error message for this run, if present.

list (*, *state*=*'DEFAULT'*, *type*=*'DEFAULT'*, *q*=*'DEFAULT'*, *permission*=*'DEFAULT'*, *hidden*=*'DEFAULT'*, *archived*=*'DEFAULT'*, *limit*=*'DEFAULT'*, *page_num*=*'DEFAULT'*, *order*=*'DEFAULT'*, *order_dir*=*'DEFAULT'*, *iterator*=*'DEFAULT'*)

List

Parameters *state* : string, optional

The job's state. One or more of queued, running, succeeded, failed, and cancelled. Specify multiple values as a comma-separated list (e.g., "A,B").

type : string, optional

The job's type. Specify multiple values as a comma-separated list (e.g., "A,B").

q : string, optional

Query string to search on the id, name, and job type

permission : string, optional

A permissions string, one of "read", "write", or "manage". Lists only jobs for which the current user has that permission.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

archived : string, optional

The archival status of the requested object(s).

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to `updated_at`. Must be one of: `updated_at`.

order_dir : string, optional

Direction in which to sort, either `asc` (ascending) or `desc` (descending) defaulting to `desc`.

iterator : bool, optional

If `True`, return a generator to iterate over all responses. Use when more results than the maximum allowed by `limit` are needed. When `True`, `limit` and `page_num` are ignored. Defaults to `False`.

Returns `id` : integer

name : string

type : string

state : string

Whether the job is idle, queued, running, cancelled, or failed.

created_at : string/date-time

updated_at : string/date-time

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

archived : string

The archival status of the requested object(s).

list_children (*id*)

Show nested tree of children that this job triggers

Parameters `id` : integer

The ID for this job.

Returns `id` : integer

name : string

type : string

state : string

created_at : string/date-time

updated_at : string/date-time

runs : list:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

children : list

list_parents (*id*)

Show chain of parents as a list that this job triggers from

Parameters **id** : integer

The ID for this job.

Returns **id** : integer

name : string

type : string

state : string

Whether the job is idle, queued, running, cancelled, or failed.

created_at : string/date-time

updated_at : string/date-time

runs : list:

```
Information about the most recent runs of the job.
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

list_projects (*id*, *, *hidden*='DEFAULT')

List the projects a Job belongs to

Parameters **id** : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns **id** : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
```

```
This user's name.
- username : string
  This user's username.
- initials : string
  This user's initials.
- online : boolean
  Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_shares (*id*)

List users and groups permissioned on this object

Parameters *id* : integer

The ID of the object.

Returns *readers* : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

post_runs (*id*)

Run a job

Parameters *id* : integer

The ID for this job.

Returns *id* : integer**state** : string**created_at** : string/time

The time that the run was queued.

started_at : string/time

The time that the run started.

finished_at : string/time

The time that the run completed.

error : string

The error message for this run, if present.

post_trigger_email (*id*)

Generate and retrieve trigger email address

Parameters *id* : integer

The ID for this job.

Returns **trigger_email** : string

Email address which may be used to trigger this job to run.

put_projects (*id*, *project_id*)

Add a Job to a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_shares_groups (*id*, *group_ids*, *permission_level*, *, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions groups has on this object

Parameters *id* : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
 send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns `readers` : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Match_Targets

`civis.resources._resources.Match_Targets`
alias of MatchTargets

Media

class `Media` (*session_kwargs*, *return_type*='civis')

Methods

<code>delete_optimizations_runs(id, run_id)</code>	Cancel a run
<code>delete_optimizations_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object

Continued on next page

Table 5.18 – continued from previous page

<i>delete_optimizations_shares_users</i> (id, user_id)	Revoke the permissions a user has on this object
<i>delete_spot_orders_shares_groups</i> (id, group_id)	Revoke the permissions a group has on this object
<i>delete_spot_orders_shares_users</i> (id, user_id)	Revoke the permissions a user has on this object
<i>get_optimizations</i> (id)	Show a single optimization
<i>get_optimizations_runs</i> (id, run_id)	Check status of a run
<i>get_spot_orders</i> (id)	Show a single spot order
<i>list_dmas</i> (*[, name, number])	List all Designated Market Areas
<i>list_optimizations</i> (*[, archived, limit, ...])	List all optimizations
<i>list_optimizations_runs</i> (id, *[, limit, ...])	List runs for the given optimization
<i>list_optimizations_runs_logs</i> (id, run_id, *)	Get the logs for a run
<i>list_optimizations_shares</i> (id)	List users and groups permissioned on this object
<i>list_ratecards</i> (*[, filename, dma_number])	List all ratecards
<i>list_spot_orders</i> (*[, id, archived])	List all spot orders
<i>list_spot_orders_shares</i> (id)	List users and groups permissioned on this object
<i>list_targets</i> (*[, name, identifier, data_source])	List all Media Targets
<i>patch_optimizations</i> (id, *[, name, runs])	Edit an existing optimization
<i>post_optimizations</i> (runs, *[, name])	Create a new optimization
<i>post_optimizations_clone</i> (id)	Clone an existing optimization
<i>post_optimizations_runs</i> (id)	Start a run
<i>post_spot_orders</i> (*[, body])	Create a spot order
<i>put_optimizations_archive</i> (id, status)	Update the archive status of this object
<i>put_optimizations_shares_groups</i> (id, ...[, ...])	Set the permissions groups has on this object
<i>put_optimizations_shares_users</i> (id, user_ids, ...)	Set the permissions users have on this object
<i>put_spot_orders</i> (id, *[, body])	Edit the specified spot order
<i>put_spot_orders_archive</i> (id, status)	Update the archive status of this object
<i>put_spot_orders_shares_groups</i> (id, group_ids, ...)	Set the permissions groups has on this object
<i>put_spot_orders_shares_users</i> (id, user_ids, ...)	Set the permissions users have on this object

delete_optimizations_runs (id, run_id)

Cancel a run

Parameters id : integer

The ID of the optimization.

run_id : integer

The ID of the run.

Returns None

Response code 202: success

delete_optimizations_shares_groups (id, group_id)

Revoke the permissions a group has on this object

Parameters id : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_optimizations_shares_users (*id, user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

delete_spot_orders_shares_groups (*id, group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_spot_orders_shares_users (*id, user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get_optimizations (*id*)

Show a single optimization

Parameters **id** : integer

The optimization ID.

Returns **id** : integer

The optimization ID.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of the optimization.

created_at : string/time

updated_at : string/time

finished_at : string/date-time

The end time of the last run.

state : string

The state of the last run.

last_run_id : integer

The ID of the last run.

spot_order_id : integer

The ID for the spot order produced by the optimization.

archived : string

The archival status of the requested object(s).

report_link : string

A link to the visual report for the optimization.

spot_order_link : string

A link to the json version of the spot order.

file_links : list

Links to the csv and xml versions of the spot order.

runs : list:

```
The runs of the optimization.
- market_id : integer
    The market ID.
- start_date : string/date
    The start date for the media run.
- end_date : string/date
    The end date for the media run.
- syscodes : list
    The syscodes for the media run.
- rate_cards : list
    The ratecards for the media run.
- constraints : list::
```

```

The constraints for the media run.
- targets : list
    The targets to constrain.
- budget : number/float
    The maximum budget for these targets.
- frequency : integer
    The maximum frequency for these targets.

```

get_optimizations_runs (*id*, *run_id*)

Check status of a run

Parameters *id* : integer

The ID of the optimization.

run_id : integer

The ID of the run.

Returns *id* : integer

The ID of the run.

optimization_id : integer

The ID of the optimization.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

get_spot_orders (*id*)

Show a single spot order

Parameters *id* : integer

The ID for the spot order.

Returns *id* : integer

The ID for the spot order.

archived : string

The archival status of the requested object(s).

csv_s3_uri : string

S3 URI for the spot order CSV file.

json_s3_uri : string

S3 URI for the spot order JSON file.

xml_archive_s3_uri : string

S3 URI for the spot order XML archive.

last_transform_job_id : integer

ID of the spot order transformation job.

list_dmas (*, *name*='DEFAULT', *number*='DEFAULT')

List all Designated Market Areas

Parameters **name** : string, optional

If specified, will be used to filter the DMAs returned. Substring matching is supported with “%” and “*” wildcards (e.g., “name=%region%” will return both “region1” and “my region”).

number : integer, optional

If specified, will be used to filter the DMAS by number.

Returns **name** : string

Name for the DMA region.

number : integer

Identifier number for a DMA.

list_optimizations (*, *archived*='DEFAULT', *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List all optimizations

Parameters **archived** : string, optional

The archival status of the requested object(s).

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to created_at. Must be one of: created_at, author, name.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The optimization ID.

author : dict:


```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

name : string

The name of the optimization.

created_at : string/time

updated_at : string/time

finished_at : string/date-time

The end time of the last run.

state : string

The state of the last run.

last_run_id : integer

The ID of the last run.

spot_order_id : integer

The ID for the spot order produced by the optimization.

archived : string

The archival status of the requested object(s).

list_optimizations_runs (*id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List runs for the given optimization

Parameters **id** : integer

The ID of the optimization.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 100.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The ID of the run.

optimization_id : integer

The ID of the optimization.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

list_optimizations_runs_logs (*id*, *run_id*, *, *last_id*=*'DEFAULT'*, *limit*=*'DEFAULT'*)

Get the logs for a run

Parameters **id** : integer

The ID of the optimization.

run_id : integer

The ID of the run.

last_id : integer, optional

The ID of the last log message received. Log entries with this ID value or lower will be omitted. Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.

limit : integer, optional

The maximum number of log messages to return. Default of 10000.

Returns **id** : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

message : string

The log message.

level : string

The level of the log. One of unknown,fatal,error,warn,info,debug.

list_optimizations_shares (*id*)

List users and groups permissioned on this object

Parameters *id* : integer

The ID of the object.

Returns *readers* : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_ratecards (*, *filename*='DEFAULT', *dma_number*='DEFAULT')

List all ratecards

Parameters *filename* : string, optional

If specified, will be used to filter the ratecards returned. Substring matching is supported with “%” and “*” wildcards (e.g., “filename=%ratecard%” will return both “ratecard 1” and “my ratecard”).

dma_number : integer, optional

If specified, will be used to filter the ratecards by DMA.

Returns *id* : integer

The ratecard ID.

filename : string

Name of the ratecard file.

start_on : string/date

First day to which the ratecard applies.

end_on : string/date

Last day to which the ratecard applies.

dma_number : integer

Number of the DMA associated with the ratecard.

list_spot_orders (*, *id*='DEFAULT', *archived*='DEFAULT')

List all spot orders

Parameters **id** : integer, optional

The ID for the spot order.

archived : string, optional

The archival status of the requested object(s).

Returns **id** : integer

The ID for the spot order.

archived : string

The archival status of the requested object(s).

list_spot_orders_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_targets (*, name='DEFAULT', identifier='DEFAULT', data_source='DEFAULT')

List all Media Targets

Parameters **name** : string, optional

The name of the target.

identifier : string, optional

A unique identifier for this target.

data_source : string, optional

The source of viewership data for this target.

Returns **name** : string

The name of the target.

identifier : string

A unique identifier for this target.

data_source : string

The source of viewership data for this target.

patch_optimizations (id, *, name='DEFAULT', runs='DEFAULT')

Edit an existing optimization

Parameters **id** : integer

The optimization ID.

name : string, optional

The name of the optimization.

runs : list, optional:

```
The runs of the optimization.
- market_id : integer
  The market ID.
- start_date : string/date
  The start date for the media run.
- end_date : string/date
  The end date for the media run.
- syscodes : list
  The syscodes for the media run.
- rate_cards : list
  The ratecards for the media run.
- constraints : list::
  The constraints for the media run.
  - targets : list
    The targets to constrain.
  - budget : number/float
```

```
The maximum budget for these targets.  
- frequency : integer  
The maximum frequency for these targets.
```

Returns `id` : integer

The optimization ID.

author : dict:

```
- id : integer  
    The ID of this user.  
- name : string  
    This user's name.  
- username : string  
    This user's username.  
- initials : string  
    This user's initials.  
- online : boolean  
    Whether this user is online.
```

name : string

The name of the optimization.

created_at : string/time

updated_at : string/time

finished_at : string/date-time

The end time of the last run.

state : string

The state of the last run.

last_run_id : integer

The ID of the last run.

spot_order_id : integer

The ID for the spot order produced by the optimization.

archived : string

The archival status of the requested object(s).

report_link : string

A link to the visual report for the optimization.

spot_order_link : string

A link to the json version of the spot order.

file_links : list

Links to the csv and xml versions of the spot order.

runs : list:

```

The runs of the optimization.
- market_id : integer
    The market ID.
- start_date : string/date
    The start date for the media run.
- end_date : string/date
    The end date for the media run.
- syscodes : list
    The syscodes for the media run.
- rate_cards : list
    The ratecards for the media run.
- constraints : list::
    The constraints for the media run.
    - targets : list
        The targets to constrain.
    - budget : number/float
        The maximum budget for these targets.
    - frequency : integer
        The maximum frequency for these targets.

```

post_optimizations (*runs*, *, *name*='DEFAULT')

Create a new optimization

Parameters *runs* : list:

```

The runs of the optimization.
- market_id : integer
    The market ID.
- start_date : string/date
    The start date for the media run.
- end_date : string/date
    The end date for the media run.
- syscodes : list
    The syscodes for the media run.
- rate_cards : list
    The ratecards for the media run.
- constraints : list::
    The constraints for the media run.
    - targets : list
        The targets to constrain.
    - budget : number/float
        The maximum budget for these targets.
    - frequency : integer
        The maximum frequency for these targets.

```

name : string, optional

The name of the optimization.

Returns *id* : integer

The optimization ID.

author : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string

```

```
This user's username.  
- initials : string  
  This user's initials.  
- online : boolean  
  Whether this user is online.
```

name : string

The name of the optimization.

created_at : string/time

updated_at : string/time

finished_at : string/date-time

The end time of the last run.

state : string

The state of the last run.

last_run_id : integer

The ID of the last run.

spot_order_id : integer

The ID for the spot order produced by the optimization.

archived : string

The archival status of the requested object(s).

report_link : string

A link to the visual report for the optimization.

spot_order_link : string

A link to the json version of the spot order.

file_links : list

Links to the csv and xml versions of the spot order.

runs : list:

```
The runs of the optimization.  
- market_id : integer  
  The market ID.  
- start_date : string/date  
  The start date for the media run.  
- end_date : string/date  
  The end date for the media run.  
- syscodes : list  
  The syscodes for the media run.  
- rate_cards : list  
  The ratecards for the media run.  
- constraints : list::  
  The constraints for the media run.  
  - targets : list  
    The targets to constrain.  
  - budget : number/float  
    The maximum budget for these targets.
```



```
- frequency : integer
    The maximum frequency for these targets.
```

post_optimizations_clone (*id*)

Clone an existing optimization

Parameters *id* : integer

The optimization ID.

Returns *id* : integer

The optimization ID.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of the optimization.

created_at : string/time**updated_at** : string/time**finished_at** : string/date-time

The end time of the last run.

state : string

The state of the last run.

last_run_id : integer

The ID of the last run.

spot_order_id : integer

The ID for the spot order produced by the optimization.

archived : string

The archival status of the requested object(s).

report_link : string

A link to the visual report for the optimization.

spot_order_link : string

A link to the json version of the spot order.

file_links : list

Links to the csv and xml versions of the spot order.

runs : list:

The runs of the optimization.

- **market_id** : integer
The market ID.
- **start_date** : string/date
The start date **for** the media run.
- **end_date** : string/date
The end date **for** the media run.
- **syscodes** : list
The syscodes **for** the media run.
- **rate_cards** : list
The ratecards **for** the media run.
- **constraints** : list::
 - **targets** : list
The targets to constrain.
 - **budget** : number/float
The maximum budget **for** these targets.
 - **frequency** : integer
The maximum frequency **for** these targets.

post_optimizations_runs (*id*)

Start a run

Parameters **id** : integer

The ID of the optimization.

Returns **id** : integer

The ID of the run.

optimization_id : integer

The ID of the optimization.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

post_spot_orders (*, *body*='DEFAULT')

Create a spot order

Parameters **body** : string, optional

CSV body of a spot order.

Returns **id** : integer

The ID for the spot order.

archived : string

The archival status of the requested object(s).

csv_s3_uri : string

S3 URI for the spot order CSV file.

json_s3_uri : string

S3 URI for the spot order JSON file.

xml_archive_s3_uri : string

S3 URI for the spot order XML archive.

last_transform_job_id : integer

ID of the spot order transformation job.

put_optimizations_archive (*id, status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **id** : integer

The optimization ID.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of the optimization.

created_at : string/time

updated_at : string/time

finished_at : string/date-time

The end time of the last run.

state : string

The state of the last run.

last_run_id : integer

The ID of the last run.

spot_order_id : integer

The ID for the spot order produced by the optimization.

archived : string

The archival status of the requested object(s).

report_link : string

A link to the visual report for the optimization.

spot_order_link : string

A link to the json version of the spot order.

file_links : list

Links to the csv and xml versions of the spot order.

runs : list:

The runs of the optimization.

- **market_id** : integer
The market ID.
- **start_date** : string/date
The start date **for** the media run.
- **end_date** : string/date
The end date **for** the media run.
- **syscodes** : list
The syscodes **for** the media run.
- **rate_cards** : list
The ratecards **for** the media run.
- **constraints** : list::
 - **targets** : list
The targets to constrain.
 - **budget** : number/float
The maximum budget **for** these targets.
 - **frequency** : integer
The maximum frequency **for** these targets.

put_optimizations_shares_groups (*id*, *group_ids*, *permission_level*,
*, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns `readers` : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_optimizations_shares_users (*id*, *user_ids*, *permission_level*,
*, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns `readers` : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_spot_orders (*id*, *, *body*='DEFAULT')

Edit the specified spot order

Parameters *id* : integer

The ID for the spot order.

body : string, optional

CSV body of a spot order.

Returns *id* : integer

The ID for the spot order.

archived : string

The archival status of the requested object(s).

csv_s3_uri : string

S3 URI for the spot order CSV file.

json_s3_uri : string

S3 URI for the spot order JSON file.

xml_archive_s3_uri : string

S3 URI for the spot order XML archive.

last_transform_job_id : integer

ID of the spot order transformation job.

put_spot_orders_archive (*id, status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **id** : integer

The ID for the spot order.

archived : string

The archival status of the requested object(s).

csv_s3_uri : string

S3 URI for the spot order CSV file.

json_s3_uri : string

S3 URI for the spot order JSON file.

xml_archive_s3_uri : string

S3 URI for the spot order XML archive.

last_transform_job_id : integer

ID of the spot order transformation job.

put_spot_orders_shares_groups (*id, group_ids, permission_level, *, share_email_body='DEFAULT', send_shared_email='DEFAULT'*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_spot_orders_shares_users (*id*, *user_ids*, *permission_level*,
*, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:


```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Models

class Models (*session_kwargs*, *return_type*='civis')

Methods

<code>delete_builds(id, build_id)</code>	Cancel a build
<code>delete_projects(id, project_id)</code>	Remove a models from a project
<code>delete_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>get(id)</code>	Retrieve model configuration
<code>get_builds(id, build_id)</code>	Check status of a build
<code>list(*[, model_name, training_table_name, ...])</code>	List
<code>list_builds(id, *[, limit, page_num, order, ...])</code>	List builds for the given model
<code>list_builds_logs(id, build_id, *[, last_id, ...])</code>	Get the logs for a build
<code>list_projects(id, *[, hidden])</code>	List the projects a models belongs to
<code>list_schedules(id)</code>	Show the model build schedule
<code>list_shares(id)</code>	List users and groups permissioned on this object

Continued on next page

Table 5.19 – continued from previous page

<i>list_types()</i>	List all available model types
<i>patch</i> (id, *, table_name, database_id, ...)	Update model configuration
<i>post</i> (*, table_name, database_id, ...)	Create new configuration for a model
<i>post_builds</i> (id)	Start a build
<i>put_archive</i> (id, status)	Update the archive status of this object
<i>put_predictions</i> (id, table_name, primary_key, *)	Add a table on which to apply the predictive model
<i>put_projects</i> (id, project_id)	Add a models to a project
<i>put_schedules</i> (id, schedule)	Schedule the model build
<i>put_shares_groups</i> (id, group_ids, ...[, ...])	Set the permissions groups has on this object
<i>put_shares_users</i> (id, user_ids, ...[, ...])	Set the permissions users have on this object

delete_builds (*id*, *build_id*)

Cancel a build

Parameters *id* : integer

The ID of the model.

build_id : integer

The ID of the build.

Returns None

Response code 202: success

delete_projects (*id*, *project_id*)

Remove a models from a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_shares_groups (*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters *id* : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users (*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters *id* : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get (*id*)

Retrieve model configuration

Parameters *id* : integer

The ID of the model.

Returns *id* : integer

The ID of the model.

table_name : string

The qualified name of the table containing the training set from which to build the model.

database_id : integer

The ID of the database holding the training set table used to build the model.

credential_id : integer

The ID of the credential used to read the target table. Defaults to the user's default credential.

model_name : string

The name of the model.

description : string

A description of the model.

interaction_terms : boolean

Whether to search for interaction terms.

box_cox_transformation : boolean

Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.

model_type_id : integer

The ID of the model's type.

primary_key : string

The unique ID (primary key) of the training dataset.

dependent_variable : string

The dependent variable of the training dataset.

dependent_variable_order : list

The order of dependent variables, especially useful for Ordinal Modeling.

excluded_columns : list

A list of columns which will be considered ineligible to be independent variables.

limiting_sql : string

A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., “id > 105”).

active_build_id : integer

The ID of the current active build, the build used to score predictions.

cross_validation_parameters : dict

Cross validation parameter grid for tree methods, e.g. {“n_estimators”: [100, 200, 500], “learning_rate”: [0.01, 0.1], “max_depth”: [2, 3]}.

number_of_folds : integer

Number of folds for cross validation. Default value is 5.

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

parent_id : integer

The ID of the parent job that will trigger this model.

running_as : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

time_zone : string

The time zone of this model.

last_run : dict:

```

- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

hidden : boolean

The hidden status of the object.

user : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

created_at : string/date-time

The time the model was created.

updated_at : string/date-time

The time the model was updated.

current_build_state : string

The status of the current model build. One of “succeeded”, “failed”, “queued”, or “running,” or “idle”, if no build has been attempted.

current_build_exception : string

Exception message, if applicable, of the current model build.

builds : list:

```
A list of trained models available for making predictions.
- id : integer
    The ID of the model build.
- name : string
    The name of the model build.
- created_at : string
    The time the model build was created.
- description : string
    A description of the model build.
- root_mean_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
- r_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
- roc_auc : number/float
    A key metric for binary, multinomial, and ordinal models. Nil
    for other
    model types.
```

predictions : list:

```
The tables upon which the model will be applied.
- id : integer
    The ID of the model to which to apply the prediction.
- table_name : string
    The qualified name of the table on which to apply the
    predictive model.
- primary_key : list
    The primary key or composite keys of the table being predicted.
- limiting_sql : string
    A SQL WHERE clause used to scope the rows to be predicted.
- output_table : string
    The qualified name of the table to be created which will
    contain the
    model's predictions.
- schedule : dict::
    - scheduled : boolean
        If the object is scheduled
    - scheduled_days : list
        Day based on numeric value starting at 0 for Sunday
    - scheduled_hours : list
        Hours of the day it is scheduled on
    - scheduled_minutes : list
        Minutes of the day it is scheduled on
    - scheduled_runs_per_hour : integer
        Alternative to scheduled minutes, number of times to run
    per hour
- state : string
    The status of the prediction. One of: "succeeded", "failed",
    "queued",
    or "running", or "idle", if no build has been attempted.
```

last_output_location : string

The output JSON for the last build.

archived : string

The archival status of the requested object(s).

get_builds (*id*, *build_id*)

Check status of a build

Parameters *id* : integer

The ID of the model.

build_id : integer

The ID of the build.

Returns *id* : integer

The ID of the model build.

state : string

The state of the model build. one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

error : string

The error, if any, returned by the build.

name : string

The name of the model build.

created_at : string

The time the model build was created.

description : string

A description of the model build.

root_mean_squared_error : number/float

A key metric for continuous models. Nil for other model types.

r_squared_error : number/float

A key metric for continuous models. Nil for other model types.

roc_auc : number/float

A key metric for binary, multinomial, and ordinal models. Nil for other model types.

transformation_metadata : string

A string representing the full JSON output of the metadata for transformation of column names

output : string

A string representing the JSON output for the specified build. Only present when smaller than 10KB in size.

output_location : string

A URL representing the location of the full JSON output for the specified build. The URL link will be valid for 5 minutes.

list (*, *model_name*='DEFAULT', *training_table_name*='DEFAULT', *dependent_variable*='DEFAULT', *author*='DEFAULT', *status*='DEFAULT', *hidden*='DEFAULT', *archived*='DEFAULT', *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List

Parameters **model_name** : string, optional

If specified, will be used to filter the models returned. Substring matching is supported. (e.g., “modelName=model” will return both “model1” and “my model”).

training_table_name : string, optional

If specified, will be used to filter the models returned by the training dataset table name. Substring matching is supported. (e.g., “trainingTableName=table” will return both “table1” and “my_table”).

dependent_variable : string, optional

If specified, will be used to filter the models returned by the dependent variable column name. Substring matching is supported. (e.g., “dependentVariable=predictor” will return both “predictor” and “my predictor”).

author : string, optional

If specified, return models from this author. It accepts a comma-separated list of author ids.

status : string, optional

If specified, returns models with one of these statuses. It accepts a comma-separated list, possible values are ‘running’, ‘failed’, ‘succeeded’, ‘idle’, ‘scheduled’.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

archived : string, optional

The archival status of the requested object(s).

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to updated_at. Must be one of: updated_at, name, created_at, last_run.updated_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The ID of the model.

table_name : string

The qualified name of the table containing the training set from which to build the model.

database_id : integer

The ID of the database holding the training set table used to build the model.

credential_id : integer

The ID of the credential used to read the target table. Defaults to the user's default credential.

model_name : string

The name of the model.

description : string

A description of the model.

interaction_terms : boolean

Whether to search for interaction terms.

box_cox_transformation : boolean

Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.

model_type_id : integer

The ID of the model's type.

primary_key : string

The unique ID (primary key) of the training dataset.

dependent_variable : string

The dependent variable of the training dataset.

dependent_variable_order : list

The order of dependent variables, especially useful for Ordinal Modeling.

excluded_columns : list

A list of columns which will be considered ineligible to be independent variables.

limiting_sql : string

A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., "id > 105").

cross_validation_parameters : dict

Cross validation parameter grid for tree methods, e.g. {"n_estimators": [100, 200, 500], "learning_rate": [0.01, 0.1], "max_depth": [2, 3]}.

number_of_folds : integer

Number of folds for cross validation. Default value is 5.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
```

```
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

parent_id : integer

The ID of the parent job that will trigger this model.

time_zone : string

The time zone of this model.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/date-time

The time the model was created.

updated_at : string/date-time

The time the model was updated.

current_build_state : string

The status of the current model build. One of “succeeded”, “failed”, “queued”, or “running,” or “idle”, if no build has been attempted.

current_build_exception : string

Exception message, if applicable, of the current model build.

builds : list:

```
A list of trained models available for making predictions.
- id : integer
    The ID of the model build.
- name : string
    The name of the model build.
- created_at : string
    The time the model build was created.
- description : string
    A description of the model build.
- root_mean_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
- r_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
- roc_auc : number/float
    A key metric for binary, multinomial, and ordinal models. Nil
    for other
    model types.
```

predictions : list:

```
The tables upon which the model will be applied.
- id : integer
    The ID of the model to which to apply the prediction.
- table_name : string
    The qualified name of the table on which to apply the
    predictive model.
- primary_key : list
    The primary key or composite keys of the table being predicted.
- limiting_sql : string
    A SQL WHERE clause used to scope the rows to be predicted.
- output_table : string
    The qualified name of the table to be created which will
    contain the
    model's predictions.
- state : string
    The status of the prediction. One of: "succeeded", "failed",
    "queued",
    or "running", or "idle", if no build has been attempted.
```

last_output_location : string

The output JSON for the last build.

archived : string

The archival status of the requested object(s).

list_builds (*id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List builds for the given model

Parameters *id* : integer

The ID of the model.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 100.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns id : integer

The ID of the model build.

state : string

The state of the model build.one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

error : string

The error, if any, returned by the build.

name : string

The name of the model build.

created_at : string

The time the model build was created.

description : string

A description of the model build.

root_mean_squared_error : number/float

A key metric for continuous models. Nil for other model types.

r_squared_error : number/float

A key metric for continuous models. Nil for other model types.

roc_auc : number/float

A key metric for binary, multinomial, and ordinal models. Nil for other model types.

transformation_metadata : string

A string representing the full JSON output of the metadata for transformation of column names

output : string

A string representing the JSON output for the specified build. Only present when smaller than 10KB in size.

output_location : string

A URL representing the location of the full JSON output for the specified build.The URL link will be valid for 5 minutes.

list_builds_logs (*id, build_id, *, last_id='DEFAULT', limit='DEFAULT'*)

Get the logs for a build

Parameters `id` : integer

The ID of the model.

build_id : integer

The ID of the build.

last_id : integer, optional

The ID of the last log message received. Log entries with this ID value or lower will be omitted. Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.

limit : integer, optional

The maximum number of log messages to return. Default of 10000.

Returns `id` : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

message : string

The log message.

level : string

The level of the log. One of unknown,fatal,error,warn,info,debug.

list_projects (*id*, *, *hidden*=*'DEFAULT'*)

List the projects a models belongs to

Parameters `id` : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns `id` : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_schedules (*id*)

Show the model build schedule

Parameters **id** : integer

The ID of the model associated with this schedule.

Returns **id** : integer

The ID of the model associated with this schedule.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

list_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_types ()

List all available model types

Returns **id** : integer

The ID of the model type.

algorithm : string

The name of the algorithm used to train the model.

dv_type : string

The type of dependent variable predicted by the model.

int_allowed : boolean

Whether this model type supports searching for interaction terms.

patch (*id*, *, *table_name*='DEFAULT', *database_id*='DEFAULT', *credential_id*='DEFAULT', *model_name*='DEFAULT', *description*='DEFAULT', *interaction_terms*='DEFAULT', *box_cox_transformation*='DEFAULT', *model_type_id*='DEFAULT', *primary_key*='DEFAULT', *dependent_variable*='DEFAULT', *dependent_variable_order*='DEFAULT', *excluded_columns*='DEFAULT', *limiting_sql*='DEFAULT', *active_build_id*='DEFAULT', *cross_validation_parameters*='DEFAULT', *number_of_folds*='DEFAULT', *notifications*='DEFAULT', *schedule*='DEFAULT', *parent_id*='DEFAULT', *time_zone*='DEFAULT')

Update model configuration

Parameters **id** : integer

The ID of the model.

table_name : string, optional

The qualified name of the table containing the training set from which to build the model.

database_id : integer, optional

The ID of the database holding the training set table used to build the model.

credential_id : integer, optional

The ID of the credential used to read the target table. Defaults to the user's default credential.

model_name : string, optional

The name of the model.

description : string, optional

A description of the model.

interaction_terms : boolean, optional

Whether to search for interaction terms.

box_cox_transformation : boolean, optional

Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.

model_type_id : integer, optional

The ID of the model's type.

primary_key : string, optional

The unique ID (primary key) of the training dataset.

dependent_variable : string, optional

The dependent variable of the training dataset.

dependent_variable_order : list, optional

The order of dependent variables, especially useful for Ordinal Modeling.

excluded_columns : list, optional

A list of columns which will be considered ineligible to be independent variables.

limiting_sql : string, optional

A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., "id > 105").

active_build_id : integer, optional

The ID of the current active build, the build used to score predictions.

cross_validation_parameters : dict, optional

Cross validation parameter grid for tree methods, e.g. {"n_estimators": [100, 200, 500], "learning_rate": [0.01, 0.1], "max_depth": [2, 3]}.

number_of_folds : integer, optional

Number of folds for cross validation. Default value is 5.

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳hour
```

parent_id : integer, optional

The ID of the parent job that will trigger this model.

time_zone : string, optional

The time zone of this model.

Returns None

Response code 204: success

```
post (*,      table_name='DEFAULT',      database_id='DEFAULT',      credential_id='DEFAULT',  
      model_name='DEFAULT',      description='DEFAULT',      interaction_terms='DEFAULT',  
      box_cox_transformation='DEFAULT', model_type_id='DEFAULT', primary_key='DEFAULT',  
      dependent_variable='DEFAULT',      dependent_variable_order='DEFAULT',      ex-  
cluded_columns='DEFAULT',      limiting_sql='DEFAULT',      active_build_id='DEFAULT',  
      cross_validation_parameters='DEFAULT',      number_of_folds='DEFAULT',      notifica-  
tions='DEFAULT',      schedule='DEFAULT',      parent_id='DEFAULT',      time_zone='DEFAULT',  
      hidden='DEFAULT')
```

Create new configuration for a model

Parameters **table_name** : string, optional

The qualified name of the table containing the training set from which to build the model.

database_id : integer, optional

The ID of the database holding the training set table used to build the model.

credential_id : integer, optional

The ID of the credential used to read the target table. Defaults to the user's default credential.

model_name : string, optional

The name of the model.

description : string, optional

A description of the model.

interaction_terms : boolean, optional

Whether to search for interaction terms.

box_cox_transformation : boolean, optional

Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.

model_type_id : integer, optional

The ID of the model's type.

primary_key : string, optional

The unique ID (primary key) of the training dataset.

dependent_variable : string, optional

The dependent variable of the training dataset.

dependent_variable_order : list, optional

The order of dependent variables, especially useful for Ordinal Modeling.

excluded_columns : list, optional

A list of columns which will be considered ineligible to be independent variables.

limiting_sql : string, optional

A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., "id > 105").

active_build_id : integer, optional

The ID of the current active build, the build used to score predictions.

cross_validation_parameters : dict, optional

Cross validation parameter grid for tree methods, e.g. {"n_estimators": [100, 200, 500], "learning_rate": [0.01, 0.1], "max_depth": [2, 3]}.

number_of_folds : integer, optional

Number of folds for cross validation. Default value is 5.

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

parent_id : integer, optional

The ID of the parent job that will trigger this model.

time_zone : string, optional

The time zone of this model.

hidden : boolean, optional

The hidden status of the object.

Returns **id** : integer

The ID of the model.

table_name : string

The qualified name of the table containing the training set from which to build the model.

database_id : integer

The ID of the database holding the training set table used to build the model.

credential_id : integer

The ID of the credential used to read the target table. Defaults to the user's default credential.

model_name : string

The name of the model.

description : string

A description of the model.

interaction_terms : boolean

Whether to search for interaction terms.

box_cox_transformation : boolean

Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.

model_type_id : integer

The ID of the model's type.

primary_key : string

The unique ID (primary key) of the training dataset.

dependent_variable : string

The dependent variable of the training dataset.

dependent_variable_order : list

The order of dependent variables, especially useful for Ordinal Modeling.

excluded_columns : list

A list of columns which will be considered ineligible to be independent variables.

limiting_sql : string

A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., "id > 105").

active_build_id : integer

The ID of the current active build, the build used to score predictions.

cross_validation_parameters : dict

Cross validation parameter grid for tree methods, e.g. {"n_estimators": [100, 200, 500], "learning_rate": [0.01, 0.1], "max_depth": [2, 3]}.

number_of_folds : integer

Number of folds for cross validation. Default value is 5.

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳hour
```

parent_id : integer

The ID of the parent job that will trigger this model.

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

time_zone : string

The time zone of this model.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/date-time

The time the model was created.

updated_at : string/date-time

The time the model was updated.

current_build_state : string

The status of the current model build. One of “succeeded”, “failed”, “queued”, or “running,” or “idle”, if no build has been attempted.

current_build_exception : string

Exception message, if applicable, of the current model build.

builds : list:

```
A list of trained models available for making predictions.
- id : integer
    The ID of the model build.
- name : string
    The name of the model build.
- created_at : string
    The time the model build was created.
- description : string
    A description of the model build.
- root_mean_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
```

```

- r_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
- roc_auc : number/float
    A key metric for binary, multinomial, and ordinal models. Nil
↳ for other
    model types.

```

predictions : list:

The tables upon which the model will be applied.

```

- id : integer
    The ID of the model to which to apply the prediction.
- table_name : string
    The qualified name of the table on which to apply the
↳ predictive model.
- primary_key : list
    The primary key or composite keys of the table being predicted.
- limiting_sql : string
    A SQL WHERE clause used to scope the rows to be predicted.
- output_table : string
    The qualified name of the table to be created which will
↳ contain the
    model's predictions.
- schedule : dict::
    - scheduled : boolean
        If the object is scheduled
    - scheduled_days : list
        Day based on numeric value starting at 0 for Sunday
    - scheduled_hours : list
        Hours of the day it is scheduled on
    - scheduled_minutes : list
        Minutes of the day it is scheduled on
    - scheduled_runs_per_hour : integer
        Alternative to scheduled minutes, number of times to run
↳ per hour
- state : string
    The status of the prediction. One of: "succeeded", "failed",
↳ "queued",
    or "running", or "idle", if no build has been attempted.

```

last_output_location : string

The output JSON for the last build.

archived : string

The archival status of the requested object(s).

post_builds (id)

Start a build

Parameters id : integer

The ID of the model.

Returns id : integer

The ID of the model build.

state : string

The state of the model build. one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

error : string

The error, if any, returned by the build.

name : string

The name of the model build.

created_at : string

The time the model build was created.

description : string

A description of the model build.

root_mean_squared_error : number/float

A key metric for continuous models. Nil for other model types.

r_squared_error : number/float

A key metric for continuous models. Nil for other model types.

roc_auc : number/float

A key metric for binary, multinomial, and ordinal models. Nil for other model types.

transformation_metadata : string

A string representing the full JSON output of the metadata for transformation of column names

output : string

A string representing the JSON output for the specified build. Only present when smaller than 10KB in size.

output_location : string

A URL representing the location of the full JSON output for the specified build. The URL link will be valid for 5 minutes.

put_archive (*id*, *status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **id** : integer

The ID of the model.

table_name : string

The qualified name of the table containing the training set from which to build the model.

database_id : integer

The ID of the database holding the training set table used to build the model.

credential_id : integer

The ID of the credential used to read the target table. Defaults to the user's default credential.

model_name : string

The name of the model.

description : string

A description of the model.

interaction_terms : boolean

Whether to search for interaction terms.

box_cox_transformation : boolean

Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.

model_type_id : integer

The ID of the model's type.

primary_key : string

The unique ID (primary key) of the training dataset.

dependent_variable : string

The dependent variable of the training dataset.

dependent_variable_order : list

The order of dependent variables, especially useful for Ordinal Modeling.

excluded_columns : list

A list of columns which will be considered ineligible to be independent variables.

limiting_sql : string

A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., "id > 105").

active_build_id : integer

The ID of the current active build, the build used to score predictions.

cross_validation_parameters : dict

Cross validation parameter grid for tree methods, e.g. {"n_estimators": [100, 200, 500], "learning_rate": [0.01, 0.1], "max_depth": [2, 3]}.

number_of_folds : integer

Number of folds for cross validation. Default value is 5.

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
```

```
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
    ↳ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
    ↳ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per,
    ↳ hour
```

parent_id : integer

The ID of the parent job that will trigger this model.

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

time_zone : string

The time zone of this model.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
```

```

- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

hidden : boolean

The hidden status of the object.

user : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

created_at : string/date-time

The time the model was created.

updated_at : string/date-time

The time the model was updated.

current_build_state : string

The status of the current model build. One of “succeeded”, “failed”, “queued”, or “running,” or “idle”, if no build has been attempted.

current_build_exception : string

Exception message, if applicable, of the current model build.

builds : list:

```

A list of trained models available for making predictions.
- id : integer
    The ID of the model build.
- name : string
    The name of the model build.
- created_at : string
    The time the model build was created.
- description : string
    A description of the model build.
- root_mean_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
- r_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
- roc_auc : number/float
    A key metric for binary, multinomial, and ordinal models. Nil
    for other model types.

```

predictions : list:

```
The tables upon which the model will be applied.
- id : integer
    The ID of the model to which to apply the prediction.
- table_name : string
    The qualified name of the table on which to apply the
    ↪predictive model.
- primary_key : list
    The primary key or composite keys of the table being predicted.
- limiting_sql : string
    A SQL WHERE clause used to scope the rows to be predicted.
- output_table : string
    The qualified name of the table to be created which will
    ↪contain the
    model's predictions.
- schedule : dict::
    - scheduled : boolean
        If the object is scheduled
    - scheduled_days : list
        Day based on numeric value starting at 0 for Sunday
    - scheduled_hours : list
        Hours of the day it is scheduled on
    - scheduled_minutes : list
        Minutes of the day it is scheduled on
    - scheduled_runs_per_hour : integer
        Alternative to scheduled minutes, number of times to run
    ↪per hour
- state : string
    The status of the prediction. One of: "succeeded", "failed",
    ↪"queued",
    or "running", or "idle", if no build has been attempted.
```

last_output_location : string

The output JSON for the last build.

archived : string

The archival status of the requested object(s).

```
put_predictions (id, table_name, primary_key, *, limiting_sql='DEFAULT', out-
put_table='DEFAULT', schedule='DEFAULT')
```

Add a table on which to apply the predictive model

Parameters **id** : integer

The ID of the model to which to apply the prediction.

table_name : string

The qualified name of the table on which to apply the predictive model.

primary_key : list

The primary key or composite keys of the table being predicted.

limiting_sql : string, optional

A SQL WHERE clause used to scope the rows to be predicted.

output_table : string, optional

The qualified name of the table to be created which will contain the model's predictions.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

Returns **id** : integer

The ID of the model to which to apply the prediction.

table_name : string

The qualified name of the table on which to apply the predictive model.

primary_key : list

The primary key or composite keys of the table being predicted.

limiting_sql : string

A SQL WHERE clause used to scope the rows to be predicted.

output_table : string

The qualified name of the table to be created which will contain the model's predictions.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

state : string

The status of the prediction. One of: “succeeded”, “failed”, “queued”, or “running,” or “idle”, if no build has been attempted.

put_projects (*id*, *project_id*)

Add a models to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_schedules (*id*, *schedule*)

Schedule the model build

Parameters *id* : integer

The ID of the model associated with this schedule.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

Returns *id* : integer

The ID of the model associated with this schedule.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

put_shares_groups (*id*, *group_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
 send_shared_email='DEFAULT')

Set the permissions groups has on this object

Parameters *id* : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns `readers` : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns `readers` : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Notebooks

class Notebooks (*session_kwargs*, *return_type*='civis')

Methods

<code>delete_deployments(notebook_id, deployment_id)</code>	Delete a Notebook deployment
<code>delete_projects(id, project_id)</code>	Remove a Notebook from a project
<code>delete_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>get(id)</code>	Get a Notebook
<code>get_deployments(notebook_id, deployment_id)</code>	Get details about a Notebook deployment
<code>list(*[, hidden, archived, author, status, ...])</code>	List Notebooks
<code>list_deployments(notebook_id, *[, ...])</code>	List deployments for a Notebook
<code>list_projects(id, *[, hidden])</code>	List the projects a Notebook belongs to
<code>list_shares(id)</code>	List users and groups permissioned on this object
<code>list_update_links(id)</code>	Get URLs to update notebook

Continued on next page

Table 5.20 – continued from previous page

<i>patch</i> (id, *[, name, language, description, ...])	Update some attributes of this Notebook
<i>post</i> (*[, name, language, description, ...])	Create a Notebook
<i>post_clone</i> (id)	Clone this notebook
<i>post_deployments</i> (notebook_id, *[, ...])	Deploy a Notebook
<i>put</i> (id, *[, name, language, description, ...])	Replace all attributes of this Notebook
<i>put_archive</i> (id, status)	Update the archive status of this object
<i>put_projects</i> (id, project_id)	Add a Notebook to a project
<i>put_shares_groups</i> (id, group_ids, ...[, ...])	Set the permissions groups has on this object
<i>put_shares_users</i> (id, user_ids, ...[, ...])	Set the permissions users have on this object

delete_deployments (*notebook_id, deployment_id*)

Delete a Notebook deployment

Parameters *notebook_id* : integer

The ID of the owning Notebook

deployment_id : integer

The ID for this deployment

Returns None

Response code 204: success

delete_projects (*id, project_id*)

Remove a Notebook from a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_shares_groups (*id, group_id*)

Revoke the permissions a group has on this object

Parameters *id* : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users (*id, user_id*)

Revoke the permissions a user has on this object

Parameters *id* : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get (*id*)

Get a Notebook

Parameters *id* : integer

Returns *id* : integer

The ID for this notebook.

name : string

The name of this notebook.

language : string

The kernel language of this notebook.

description : string

The description of this notebook.

notebook_url : string

Time-limited URL to get the .ipynb file for this notebook.

notebook_preview_url : string

Time-limited URL to get the .htm preview file for this notebook.

requirements_url : string

Time-limited URL to get the requirements.txt file for this notebook.

file_id : string

The file ID for the S3 file containing the .ipynb file.

requirements_file_id : string

The file ID for the S3 file containing the requirements.txt file.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

memory : integer

The amount of memory allocated to the notebook.

cpu : integer

The amount of cpu allocated to the the notebook.

created_at : string/time

updated_at : string/time

most_recent_deployment : dict:

```
- deployment_id : integer
    The ID for this deployment.
- user_id : integer
    The ID of the owner.
- host : string
    Domain of the deployment.
- name : string
    Name of the deployment.
- docker_image_name : string
    The name of the docker image to pull from DockerHub.
- docker_image_tag : string
    The tag of the docker image to pull from DockerHub (default:↵
↵latest).
- display_url : string
    URL that displays the deployed platform object and includes↵
↵auth token.
- memory : integer
    The memory allocated to the deployment.
- cpu : integer
    The cpu allocated to the deployment.
- state : string
    The state of the deployment.
- state_message : string
    A detailed description of the state.
- created_at : string/time
- updated_at : string/time
- published : boolean
- notebook_id : integer
    The ID of owning Notebook
```

credentials : list

A list of credential IDs to pass to the notebook.

git_url : string

The URL of the repository that will be cloned.

git_branch : string

The name of the branch that will be cloned.

git_file : string

The name of the notebook file within a repository that will be included in the deployment.

environment_variables : dict

Environment variables to be passed into the Notebook.

idle_timeout : integer

How long the notebook will stay alive without any kernel activity.

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

get_deployments (*notebook_id*, *deployment_id*)

Get details about a Notebook deployment

Parameters **notebook_id** : integer

The ID of the owning Notebook

deployment_id : integer

The ID for this deployment

Returns **deployment_id** : integer

The ID for this deployment.

user_id : integer

The ID of the owner.

host : string

Domain of the deployment.

name : string

Name of the deployment.

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

display_url : string

URL that displays the deployed platform object and includes auth token.

memory : integer

The memory allocated to the deployment.

cpu : integer

The cpu allocated to the deployment.

state : string

The state of the deployment.

state_message : string

A detailed description of the state.

created_at : string/time

updated_at : string/time

published : boolean

notebook_id : integer

The ID of owning Notebook

```
list (*, hidden='DEFAULT', archived='DEFAULT', author='DEFAULT', status='DEFAULT',
      limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', itera-
      tor='DEFAULT')
List Notebooks
```

Parameters **hidden** : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

archived : string, optional

The archival status of the requested object(s).

author : string, optional

If specified, return imports from this author. It accepts a comma-separated list of author IDs.

status : string, optional

If specified, returns notebooks with one of these statuses. It accepts a comma-separated list, possible values are 'running', 'pending', 'idle'.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to `updated_at`. Must be one of: `updated_at`, `name`, `created_at`.

order_dir : string, optional

Direction in which to sort, either `asc` (ascending) or `desc` (descending) defaulting to `desc`.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The ID for this notebook.

name : string

The name of this notebook.

language : string

The kernel language of this notebook.

description : string

The description of this notebook.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

most_recent_deployment : dict:

```
- deployment_id : integer
    The ID for this deployment.
- user_id : integer
    The ID of the owner.
- host : string
    Domain of the deployment.
- name : string
    Name of the deployment.
- docker_image_name : string
    The name of the docker image to pull from DockerHub.
- docker_image_tag : string
    The tag of the docker image to pull from DockerHub (default: latest).
- memory : integer
    The memory allocated to the deployment.
- cpu : integer
    The cpu allocated to the deployment.
- state : string
    The state of the deployment.
- state_message : string
    A detailed description of the state.
- created_at : string/time
- updated_at : string/time
- published : boolean
- notebook_id : integer
    The ID of owning Notebook
```

archived : string

The archival status of the requested object(s).

list_deployments (*notebook_id*, *, *deployment_id*='DEFAULT', *limit*='DEFAULT',
 page_num='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List deployments for a Notebook

Parameters **notebook_id** : integer

The ID of the owning Notebook

deployment_id : integer, optional

The ID for this deployment

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to created_at. Must be one of: created_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns deployment_id : integer

The ID for this deployment.

user_id : integer

The ID of the owner.

host : string

Domain of the deployment.

name : string

Name of the deployment.

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

memory : integer

The memory allocated to the deployment.

cpu : integer

The cpu allocated to the deployment.

state : string

The state of the deployment.

state_message : string

A detailed description of the state.

created_at : string/time

updated_at : string/time

published : boolean

notebook_id : integer

The ID of owning Notebook

list_projects (*id*, *, *hidden*=*'DEFAULT'*)

List the projects a Notebook belongs to

Parameters **id** : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns **id** : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_shares (*id*)

List users and groups permissioned on this object

Parameters *id* : integer

The ID of the object.

Returns *readers* : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_update_links (*id*)

Get URLs to update notebook

Parameters *id* : integer

Returns *update_url* : string

Time-limited URL to PUT new contents of the .ipynb file for this notebook.

update_preview_url : string

Time-limited URL to PUT new contents of the .htm preview file for this notebook.

patch (*id*, *, *name*='DEFAULT', *language*='DEFAULT', *description*='DEFAULT', *file_id*='DEFAULT', *requirements_file_id*='DEFAULT', *requirements*='DEFAULT', *docker_image_name*='DEFAULT', *docker_image_tag*='DEFAULT', *memory*='DEFAULT', *cpu*='DEFAULT', *credentials*='DEFAULT', *git_url*='DEFAULT', *git_branch*='DEFAULT', *git_file*='DEFAULT', *environment_variables*='DEFAULT', *idle_timeout*='DEFAULT')

Update some attributes of this Notebook

Parameters **id** : integer

The ID for this notebook.

name : string, optional

The name of this notebook.

language : string, optional

The kernel language of this notebook.

description : string, optional

The description of this notebook.

file_id : string, optional

The file ID for the S3 file containing the .ipynb file.

requirements_file_id : string, optional

The file ID for the S3 file containing the requirements.txt file.

requirements : string, optional

The requirements txt file.

docker_image_name : string, optional

The name of the docker image to pull from DockerHub.

docker_image_tag : string, optional

The tag of the docker image to pull from DockerHub (default: latest).

memory : integer, optional

The amount of memory allocated to the notebook.

cpu : integer, optional

The amount of cpu allocated to the the notebook.

credentials : list, optional

A list of credential IDs to pass to the notebook.

git_url : string, optional

The URL of the repository that will be cloned.

git_branch : string, optional

The name of the branch that will be cloned.

git_file : string, optional

The name of the notebook file within a repository that will be included in the deployment.

environment_variables : dict, optional

Environment variables to be passed into the Notebook.

idle_timeout : integer, optional

How long the notebook will stay alive without any kernel activity.

Returns **id** : integer

The ID for this notebook.

name : string

The name of this notebook.

language : string

The kernel language of this notebook.

description : string

The description of this notebook.

notebook_url : string

Time-limited URL to get the .ipynb file for this notebook.

notebook_preview_url : string

Time-limited URL to get the .htm preview file for this notebook.

requirements_url : string

Time-limited URL to get the requirements.txt file for this notebook.

file_id : string

The file ID for the S3 file containing the .ipynb file.

requirements_file_id : string

The file ID for the S3 file containing the requirements.txt file.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

memory : integer

The amount of memory allocated to the notebook.

cpu : integer

The amount of cpu allocated to the the notebook.

created_at : string/time

updated_at : string/time

most_recent_deployment : dict:

```
- deployment_id : integer
    The ID for this deployment.
- user_id : integer
    The ID of the owner.
- host : string
    Domain of the deployment.
- name : string
    Name of the deployment.
- docker_image_name : string
    The name of the docker image to pull from DockerHub.
- docker_image_tag : string
    The tag of the docker image to pull from DockerHub (default:↵
↵latest).
- display_url : string
    URL that displays the deployed platform object and includes↵
↵auth token.
- memory : integer
    The memory allocated to the deployment.
- cpu : integer
    The cpu allocated to the deployment.
- state : string
    The state of the deployment.
- state_message : string
    A detailed description of the state.
- created_at : string/time
- updated_at : string/time
- published : boolean
- notebook_id : integer
    The ID of owning Notebook
```

credentials : list

A list of credential IDs to pass to the notebook.

git_url : string

The URL of the repository that will be cloned.

git_branch : string

The name of the branch that will be cloned.

git_file : string

The name of the notebook file within a repository that will be included in the deployment.

environment_variables : dict

Environment variables to be passed into the Notebook.

idle_timeout : integer

How long the notebook will stay alive without any kernel activity.

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

post (*, name='DEFAULT', language='DEFAULT', description='DEFAULT', file_id='DEFAULT', requirements_file_id='DEFAULT', requirements='DEFAULT', docker_image_name='DEFAULT', docker_image_tag='DEFAULT', memory='DEFAULT', cpu='DEFAULT', credentials='DEFAULT', git_url='DEFAULT', git_branch='DEFAULT', git_file='DEFAULT', environment_variables='DEFAULT', idle_timeout='DEFAULT', hidden='DEFAULT')

Create a Notebook

Parameters **name** : string, optional

The name of this notebook.

language : string, optional

The kernel language of this notebook.

description : string, optional

The description of this notebook.

file_id : string, optional

The file ID for the S3 file containing the .ipynb file.

requirements_file_id : string, optional

The file ID for the S3 file containing the requirements.txt file.

requirements : string, optional

The requirements txt file.

docker_image_name : string, optional

The name of the docker image to pull from DockerHub.

docker_image_tag : string, optional

The tag of the docker image to pull from DockerHub (default: latest).

memory : integer, optional

The amount of memory allocated to the notebook.

cpu : integer, optional

The amount of cpu allocated to the the notebook.

credentials : list, optional

A list of credential IDs to pass to the notebook.

git_url : string, optional

The URL of the repository that will be cloned.

git_branch : string, optional

The name of the branch that will be cloned.

git_file : string, optional

The name of the notebook file within a repository that will be included in the deployment.

environment_variables : dict, optional

Environment variables to be passed into the Notebook.

idle_timeout : integer, optional

How long the notebook will stay alive without any kernel activity.

hidden : boolean, optional

The hidden status of the object.

Returns **id** : integer

The ID for this notebook.

name : string

The name of this notebook.

language : string

The kernel language of this notebook.

description : string

The description of this notebook.

notebook_url : string

Time-limited URL to get the .ipynb file for this notebook.

notebook_preview_url : string

Time-limited URL to get the .htm preview file for this notebook.

requirements_url : string

Time-limited URL to get the requirements.txt file for this notebook.

file_id : string

The file ID for the S3 file containing the .ipynb file.

requirements_file_id : string

The file ID for the S3 file containing the requirements.txt file.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

memory : integer

The amount of memory allocated to the notebook.

cpu : integer

The amount of cpu allocated to the the notebook.

created_at : string/time

updated_at : string/time

most_recent_deployment : dict:

```
- deployment_id : integer
    The ID for this deployment.
- user_id : integer
    The ID of the owner.
- host : string
    Domain of the deployment.
- name : string
    Name of the deployment.
- docker_image_name : string
    The name of the docker image to pull from DockerHub.
- docker_image_tag : string
    The tag of the docker image to pull from DockerHub (default: latest).
- display_url : string
    URL that displays the deployed platform object and includes auth token.
- memory : integer
    The memory allocated to the deployment.
- cpu : integer
    The cpu allocated to the deployment.
- state : string
    The state of the deployment.
- state_message : string
    A detailed description of the state.
- created_at : string/time
- updated_at : string/time
- published : boolean
- notebook_id : integer
    The ID of owning Notebook
```

credentials : list

A list of credential IDs to pass to the notebook.

git_url : string

The URL of the repository that will be cloned.

git_branch : string

The name of the branch that will be cloned.

git_file : string

The name of the notebook file within a repository that will be included in the deployment.

environment_variables : dict

Environment variables to be passed into the Notebook.

idle_timeout : integer

How long the notebook will stay alive without any kernel activity.

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

post_clone (*id*)

Clone this notebook

Parameters *id* : integer

Returns *id* : integer

The ID for this notebook.

name : string

The name of this notebook.

language : string

The kernel language of this notebook.

description : string

The description of this notebook.

notebook_url : string

Time-limited URL to get the .ipynb file for this notebook.

notebook_preview_url : string

Time-limited URL to get the .htm preview file for this notebook.

requirements_url : string

Time-limited URL to get the requirements.txt file for this notebook.

file_id : string

The file ID for the S3 file containing the .ipynb file.

requirements_file_id : string

The file ID for the S3 file containing the requirements.txt file.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

memory : integer

The amount of memory allocated to the notebook.

cpu : integer

The amount of cpu allocated to the the notebook.

created_at : string/time

updated_at : string/time

most_recent_deployment : dict:

```
- deployment_id : integer
    The ID for this deployment.
- user_id : integer
    The ID of the owner.
- host : string
    Domain of the deployment.
- name : string
    Name of the deployment.
- docker_image_name : string
    The name of the docker image to pull from DockerHub.
- docker_image_tag : string
    The tag of the docker image to pull from DockerHub (default:
↳latest).
- display_url : string
    URL that displays the deployed platform object and includes
↳auth token.
- memory : integer
    The memory allocated to the deployment.
- cpu : integer
    The cpu allocated to the deployment.
- state : string
    The state of the deployment.
- state_message : string
    A detailed description of the state.
- created_at : string/time
- updated_at : string/time
- published : boolean
- notebook_id : integer
    The ID of owning Notebook
```

credentials : list

A list of credential IDs to pass to the notebook.

git_url : string

The URL of the repository that will be cloned.

git_branch : string

The name of the branch that will be cloned.

git_file : string

The name of the notebook file within a repository that will be included in the deployment.

environment_variables : dict

Environment variables to be passed into the Notebook.

idle_timeout : integer

How long the notebook will stay alive without any kernel activity.

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

post_deployments (*notebook_id*, *, *deployment_id*=*'DEFAULT'*, *published*=*'DEFAULT'*)

Deploy a Notebook

Parameters **notebook_id** : integer

The ID of the owning Notebook

deployment_id : integer, optional

The ID for this deployment

published : boolean, optional

Returns **deployment_id** : integer

The ID for this deployment.

user_id : integer

The ID of the owner.

host : string

Domain of the deployment.

name : string

Name of the deployment.

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

display_url : string

URL that displays the deployed platform object and includes auth token.

memory : integer

The memory allocated to the deployment.

cpu : integer

The cpu allocated to the deployment.

state : string

The state of the deployment.

state_message : string

A detailed description of the state.

created_at : string/time

updated_at : string/time

published : boolean

notebook_id : integer

The ID of owning Notebook

```
put (id, *, name='DEFAULT', language='DEFAULT', description='DEFAULT', file_id='DEFAULT',
      requirements_file_id='DEFAULT', requirements='DEFAULT', docker_image_name='DEFAULT',
      docker_image_tag='DEFAULT', memory='DEFAULT', cpu='DEFAULT', creden-
      tials='DEFAULT', git_url='DEFAULT', git_branch='DEFAULT', git_file='DEFAULT', envi-
      ronment_variables='DEFAULT', idle_timeout='DEFAULT')
Replace all attributes of this Notebook
```

Parameters

id : integer

The ID for this notebook.

name : string, optional

The name of this notebook.

language : string, optional

The kernel language of this notebook.

description : string, optional

The description of this notebook.

file_id : string, optional

The file ID for the S3 file containing the .ipynb file.

requirements_file_id : string, optional

The file ID for the S3 file containing the requirements.txt file.

requirements : string, optional

The requirements txt file.

docker_image_name : string, optional

The name of the docker image to pull from DockerHub.

docker_image_tag : string, optional

The tag of the docker image to pull from DockerHub (default: latest).

memory : integer, optional

The amount of memory allocated to the notebook.

cpu : integer, optional

The amount of cpu allocated to the the notebook.

credentials : list, optional

A list of credential IDs to pass to the notebook.

git_url : string, optional

The URL of the repository that will be cloned.

git_branch : string, optional

The name of the branch that will be cloned.

git_file : string, optional

The name of the notebook file within a repository that will be included in the deployment.

environment_variables : dict, optional

Environment variables to be passed into the Notebook.

idle_timeout : integer, optional

How long the notebook will stay alive without any kernel activity.

Returns id : integer

The ID for this notebook.

name : string

The name of this notebook.

language : string

The kernel language of this notebook.

description : string

The description of this notebook.

notebook_url : string

Time-limited URL to get the .ipynb file for this notebook.

notebook_preview_url : string

Time-limited URL to get the .htm preview file for this notebook.

requirements_url : string

Time-limited URL to get the requirements.txt file for this notebook.

file_id : string

The file ID for the S3 file containing the .ipynb file.

requirements_file_id : string

The file ID for the S3 file containing the requirements.txt file.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

memory : integer

The amount of memory allocated to the notebook.

cpu : integer

The amount of cpu allocated to the the notebook.

created_at : string/time

updated_at : string/time

most_recent_deployment : dict:

```
- deployment_id : integer
    The ID for this deployment.
- user_id : integer
    The ID of the owner.
- host : string
    Domain of the deployment.
- name : string
    Name of the deployment.
- docker_image_name : string
    The name of the docker image to pull from DockerHub.
- docker_image_tag : string
    The tag of the docker image to pull from DockerHub (default:
↳latest).
- display_url : string
    URL that displays the deployed platform object and includes
↳auth token.
- memory : integer
    The memory allocated to the deployment.
- cpu : integer
    The cpu allocated to the deployment.
- state : string
    The state of the deployment.
- state_message : string
    A detailed description of the state.
- created_at : string/time
- updated_at : string/time
- published : boolean
- notebook_id : integer
    The ID of owning Notebook
```

credentials : list

A list of credential IDs to pass to the notebook.

git_url : string

The URL of the repository that will be cloned.

git_branch : string

The name of the branch that will be cloned.

git_file : string

The name of the notebook file within a repository that will be included in the deployment.

environment_variables : dict

Environment variables to be passed into the Notebook.

idle_timeout : integer

How long the notebook will stay alive without any kernel activity.

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

put_archive (*id, status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **id** : integer

The ID for this notebook.

name : string

The name of this notebook.

language : string

The kernel language of this notebook.

description : string

The description of this notebook.

notebook_url : string

Time-limited URL to get the .ipynb file for this notebook.

notebook_preview_url : string

Time-limited URL to get the .htm preview file for this notebook.

requirements_url : string

Time-limited URL to get the requirements.txt file for this notebook.

file_id : string

The file ID for the S3 file containing the .ipynb file.

requirements_file_id : string

The file ID for the S3 file containing the requirements.txt file.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
```

```

    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

memory : integer

The amount of memory allocated to the notebook.

cpu : integer

The amount of cpu allocated to the the notebook.

created_at : string/time

updated_at : string/time

most_recent_deployment : dict:

```

- deployment_id : integer
    The ID for this deployment.
- user_id : integer
    The ID of the owner.
- host : string
    Domain of the deployment.
- name : string
    Name of the deployment.
- docker_image_name : string
    The name of the docker image to pull from DockerHub.
- docker_image_tag : string
    The tag of the docker image to pull from DockerHub (default: latest).
- display_url : string
    URL that displays the deployed platform object and includes auth token.
- memory : integer
    The memory allocated to the deployment.
- cpu : integer
    The cpu allocated to the deployment.
- state : string
    The state of the deployment.
- state_message : string
    A detailed description of the state.
- created_at : string/time
- updated_at : string/time
- published : boolean
- notebook_id : integer
    The ID of owning Notebook

```

credentials : list

A list of credential IDs to pass to the notebook.

git_url : string

The URL of the repository that will be cloned.

git_branch : string

The name of the branch that will be cloned.

git_file : string

The name of the notebook file within a repository that will be included in the deployment.

environment_variables : dict

Environment variables to be passed into the Notebook.

idle_timeout : integer

How long the notebook will stay alive without any kernel activity.

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

put_projects (*id*, *project_id*)

Add a Notebook to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_shares_groups (*id*, *group_ids*, *permission_level*, *, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:


```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::  
  - id : integer  
  - name : string  
- groups : list::  
  - id : integer  
  - name : string
```

writers : dict:

```
- users : list::  
  - id : integer  
  - name : string  
- groups : list::  
  - id : integer  
  - name : string
```

owners : dict:

```
- users : list::  
  - id : integer  
  - name : string  
- groups : list::  
  - id : integer  
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Notifications

class Notifications (*session_kwargs*, *return_type*='civis')

Methods

<code>list(*[, last_event_id, r, mock])</code>	Receive a stream of notifications as they come in
--	---

list (*, *last_event_id*='DEFAULT', *r*='DEFAULT', *mock*='DEFAULT')

Receive a stream of notifications as they come in

Parameters **last_event_id** : string, optional

allows browser to keep track of last event fired

r : string, optional

specifies retry/reconnect timeout

mock : string, optional

used for testing

Returns None

Response code 200: success

Ontology

```
class Ontology (session_kwargs, return_type='civis')
```

Methods

<code>list(*[, subset])</code>	List the ontology of column names Civis uses
--------------------------------	--

```
list (*, subset='DEFAULT')
```

List the ontology of column names Civis uses

Parameters `subset` : string, optional

A subset of fields to return.

Returns `key` : string

title : string

desc : string

A description of this field.

aliases : list

Predictions

```
class Predictions (session_kwargs, return_type='civis')
```

Methods

<code>delete_runs(id, run_id)</code>	Cancel a run
<code>get(id)</code>	Show the specified prediction
<code>get_runs(id, run_id)</code>	Check status of a run
<code>list(*[, model_id])</code>	List predictions
<code>list_runs(id, *[, limit, page_num, order, ...])</code>	List runs for the given prediction
<code>list_runs_logs(id, run_id, *[, last_id, limit])</code>	Get the logs for a run
<code>list_schedules(id)</code>	Show the prediction schedule
<code>patch(id, *[, output_table_name, ...])</code>	Update a prediction
<code>post_runs(id)</code>	Start a run
<code>put_schedules(id, *[, schedule, ...])</code>	Schedule the prediction

```
delete_runs (id, run_id)
```

Cancel a run

Parameters `id` : integer

The ID of the prediction.

run_id : integer

The ID of the run.

Returns None

Response code 202: success

get (*id*)

Show the specified prediction

Parameters *id* : integer

The ID of the prediction.

Returns *id* : integer

The ID of the prediction.

model_id : integer

The ID of the model used for this prediction.

scored_table_id : integer

The ID of the source table for this prediction.

scored_table_name : string

The name of the source table for this prediction.

output_table_name : string

The name of the output table for this prediction.

state : string

The state of the last run of this prediction.

error : string

The error, if any, of the last run of this prediction.

started_at : string/date-time

The start time of the last run of this prediction.

finished_at : string/date-time

The end time of the last run of this prediction.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

scored_tables : list:

```
An array of created prediction tables.
- id : integer
  The ID of the table with created predictions.
```

```

- schema : string
    The schema of table with created predictions.
- name : string
    The name of table with created predictions.
- created_at : string/date-time
    The time when the table with created predictions was created.
- score_stats : list::
    An array of metrics on the created predictions.
    - score_name : string
        The name of the score.
    - histogram : list
        The histogram of the distribution of scores.
    - avg_score : number/float
        The average score.
    - min_score : number/float
        The minimum score.
    - max_score : number/float
        The maximum score.

```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour

```

limiting_sql : string

A SQL WHERE clause used to scope the rows to be predicted.

primary_key : list

The primary key or composite keys of the table being predicted.

get_runs (*id*, *run_id*)

Check status of a run

Parameters *id* : integer

The ID of the prediction.

run_id : integer

The ID of the run.

Returns *id* : integer

The ID of the prediction run.

prediction_id : integer

The ID of the prediction.

state : string

The state of the prediction run.

exception : string

The exception, if any, returned by the prediction run.

name : string

The name of table created by this predictions run.

created_at : string/date-time

The time when the table with created predictions was created.

score_stats : list:

An array of metrics on the created predictions.

- **score_name** : string
The name of the score.
- **histogram** : list
The histogram of the distribution of scores.
- **avg_score** : number/float
The average score.
- **min_score** : number/float
The minimum score.
- **max_score** : number/float
The maximum score.

list (*, model_id='DEFAULT')

List predictions

Parameters **model_id** : integer, optional

If specified, only return predictions associated with this model ID.

Returns **id** : integer

The ID of the prediction.

model_id : integer

The ID of the model used for this prediction.

scored_table_id : integer

The ID of the source table for this prediction.

scored_table_name : string

The name of the source table for this prediction.

output_table_name : string

The name of the output table for this prediction.

state : string

The state of the last run of this prediction.

error : string

The error, if any, of the last run of this prediction.

started_at : string/date-time

The start time of the last run of this prediction.

finished_at : string/date-time

The end time of the last run of this prediction.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

list_runs (*id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List runs for the given prediction

Parameters **id** : integer

The ID of the prediction.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 100.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The ID of the prediction run.

prediction_id : integer

The ID of the prediction.

state : string

The state of the prediction run.

exception : string

The exception, if any, returned by the prediction run.

name : string

The name of table created by this predictions run.

created_at : string/date-time

The time when the table with created predictions was created.

score_stats : list:

An array of metrics on the created predictions.

- **score_name** : string
The name of the score.
- **histogram** : list
The histogram of the distribution of scores.
- **avg_score** : number/float
The average score.
- **min_score** : number/float
The minimum score.
- **max_score** : number/float
The maximum score.

list_runs_logs (*id*, *run_id*, *, *last_id*=*'DEFAULT'*, *limit*=*'DEFAULT'*)

Get the logs for a run

Parameters **id** : integer

The ID of the prediction.

run_id : integer

The ID of the run.

last_id : integer, optional

The ID of the last log message received. Log entries with this ID value or lower will be omitted. Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.

limit : integer, optional

The maximum number of log messages to return. Default of 10000.

Returns **id** : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

message : string

The log message.

level : string

The level of the log. One of unknown,fatal,error,warn,info,debug.

list_schedules (*id*)

Show the prediction schedule

Parameters **id** : integer

ID of the prediction associated with this schedule.

Returns **id** : integer

ID of the prediction associated with this schedule.

schedule : dict:


```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour

```

score_on_model_build : boolean

Whether the prediction will run after a rebuild of the associated model.

patch (*id*, *, *output_table_name*='DEFAULT', *limiting_sql*='DEFAULT', *primary_key*='DEFAULT')

Update a prediction

Parameters *id* : integer

The ID of the prediction.

output_table_name : string, optional

The name of the output table for this prediction.

limiting_sql : string, optional

A SQL WHERE clause used to scope the rows to be predicted.

primary_key : list, optional

The primary key or composite keys of the table being predicted.

Returns *id* : integer

The ID of the prediction.

model_id : integer

The ID of the model used for this prediction.

scored_table_id : integer

The ID of the source table for this prediction.

scored_table_name : string

The name of the source table for this prediction.

output_table_name : string

The name of the output table for this prediction.

state : string

The state of the last run of this prediction.

error : string

The error, if any, of the last run of this prediction.

started_at : string/date-time

The start time of the last run of this prediction.

finished_at : string/date-time

The end time of the last run of this prediction.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

scored_tables : list:

```
An array of created prediction tables.
- id : integer
    The ID of the table with created predictions.
- schema : string
    The schema of table with created predictions.
- name : string
    The name of table with created predictions.
- created_at : string/date-time
    The time when the table with created predictions was created.
- score_stats : list::
    An array of metrics on the created predictions.
    - score_name : string
        The name of the score.
    - histogram : list
        The histogram of the distribution of scores.
    - avg_score : number/float
        The average score.
    - min_score : number/float
        The minimum score.
    - max_score : number/float
        The maximum score.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

limiting_sql : string

A SQL WHERE clause used to scope the rows to be predicted.

primary_key : list

The primary key or composite keys of the table being predicted.

post_runs (*id*)

Start a run

Parameters *id* : integer

The ID of the prediction.

Returns *id* : integer

The ID of the prediction run.

prediction_id : integer

The ID of the prediction.

state : string

The state of the prediction run.

exception : string

The exception, if any, returned by the prediction run.

name : string

The name of table created by this predictions run.

created_at : string/date-time

The time when the table with created predictions was created.

score_stats : list:

An array of metrics on the created predictions.

- **score_name** : string
The name of the score.
- **histogram** : list
The histogram of the distribution of scores.
- **avg_score** : number/float
The average score.
- **min_score** : number/float
The minimum score.
- **max_score** : number/float
The maximum score.

put_schedules (*id*, *, *schedule*='DEFAULT', *score_on_model_build*='DEFAULT')

Schedule the prediction

Parameters *id* : integer

ID of the prediction associated with this schedule.

schedule : dict, optional:

- **scheduled** : boolean
If the **object is** scheduled
- **scheduled_days** : list
Day based on numeric value starting at 0 **for** Sunday
- **scheduled_hours** : list
Hours of the day it **is** scheduled on
- **scheduled_minutes** : list
Minutes of the day it **is** scheduled on
- **scheduled_runs_per_hour** : integer
Alternative to scheduled minutes, number of times to run per **hour**

score_on_model_build : boolean, optional

Whether the prediction will run after a rebuild of the associated model.

Returns **id** : integer

ID of the prediction associated with this schedule.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
    ↪hour
```

score_on_model_build : boolean

Whether the prediction will run after a rebuild of the associated model.

Projects

class Projects (*session_kwargs*, *return_type*='civis')

Methods

<code>delete_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>get(project_id)</code>	Get a detailed view of a project and the objects in it
<code>list(*[, author, permission, hidden, ...])</code>	List projects
<code>list_shares(id)</code>	List users and groups permissioned on this object
<code>post(name, description, *[, note, hidden])</code>	Create a project
<code>put(project_id, *[, name, description, note])</code>	Update a project
<code>put_archive(id, status)</code>	Update the archive status of this object
<code>put_shares_groups(id, group_ids, ...[, ...])</code>	Set the permissions groups has on this object
<code>put_shares_users(id, user_ids, ...[, ...])</code>	Set the permissions users have on this object

delete_shares_groups (*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users (*id, user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get (*project_id*)

Get a detailed view of a project and the objects in it

Parameters **project_id** : integer

Returns **id** : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

tables : list:

```
- schema : string
- name : string
- row_count : integer
- column_count : integer
- created_at : string/time
- updated_at : string/time
```

surveys : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
```

scripts : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- type : string
- finished_at : string/time
- state : string
```

imports : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- type : string
- finished_at : string/time
- state : string
```

models : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- state : string
```

notebooks : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- current_deployment_id : integer
```

workflows : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- state : string
```

reports : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- state : string
```

script_templates : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
```

files : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- file_name : string
- file_size : integer
- expired : boolean
```

app_instances : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- slug : string
```

all_objects : list:

```
- project_id : integer
- object_id : integer
- object_type : string
- fco_type : string
- sub_type : string
- name : string
- icon : string
- author : string
- archived : string
    The archival status of the requested object(s).
```

note : string

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

list (*, *author*='DEFAULT', *permission*='DEFAULT', *hidden*='DEFAULT', *archived*='DEFAULT', *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List projects

Parameters **author** : string, optional

If specified, return projects owned by this author. It accepts a comma- separated list of author ids.

permission : string, optional

A permissions string, one of “read”, “write”, or “manage”. Lists only projects for which the current user has that permission.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

archived : string, optional

The archival status of the requested object(s).

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 1000.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to `updated_at`. Must be one of: `updated_at`, `name`, `created_at`.

order_dir : string, optional

Direction in which to sort, either `asc` (ascending) or `desc` (descending) defaulting to `desc`.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by `limit` are needed. When True, `limit` and `page_num` are ignored. Defaults to False.

Returns **id** : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
```



```

    This user's username.
-   initials : string
    This user's initials.
-   online : boolean
    Whether this user is online.

```

name : string

The name of this project.

description : string

A description of the project

users : list:

```

Users who can see the project
-   id : integer
    The ID of this user.
-   name : string
    This user's name.
-   username : string
    This user's username.
-   initials : string
    This user's initials.
-   online : boolean
    Whether this user is online.

```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_shares (*id*)

List users and groups permissioned on this object

Parameters `id` : integer

The ID of the object.

Returns `readers` : dict:

```

-   users : list::
    -   id : integer
    -   name : string
-   groups : list::
    -   id : integer
    -   name : string

```

writers : dict:

```

-   users : list::
    -   id : integer
    -   name : string
-   groups : list::
    -   id : integer
    -   name : string

```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

post (*name*, *description*, *, *note*=*'DEFAULT'*, *hidden*=*'DEFAULT'*)

Create a project

Parameters **name** : string

The name of this project.

description : string

A description of the project

note : string, optional

Notes for the project

hidden : boolean, optional

The hidden status of the object.

Returns **id** : integer

The ID for this project.

author : dict:

```
- id : integer
  The ID of this user.
- name : string
  This user's name.
- username : string
  This user's username.
- initials : string
  This user's initials.
- online : boolean
  Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```

Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

auto_share : boolean

created_at : string/time

updated_at : string/time

tables : list:

```

- schema : string
- name : string
- row_count : integer
- column_count : integer
- created_at : string/time
- updated_at : string/time

```

surveys : list:

```

- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time

```

scripts : list:

```

- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- type : string
- finished_at : string/time
- state : string

```

imports : list:

```

- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- type : string
- finished_at : string/time
- state : string

```

models : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- state : string
```

notebooks : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- current_deployment_id : integer
```

workflows : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- state : string
```

reports : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- state : string
```

script_templates : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
```

files : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- file_name : string
- file_size : integer
- expired : boolean
```

app_instances : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
```

```
- name : string
- slug : string
```

all_objects : list:

```
- project_id : integer
- object_id : integer
- object_type : string
- fco_type : string
- sub_type : string
- name : string
- icon : string
- author : string
- archived : string
    The archival status of the requested object(s).
```

note : string

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

put (*project_id*, *, *name*='DEFAULT', *description*='DEFAULT', *note*='DEFAULT')

Update a project

Parameters **project_id** : integer

name : string, optional

The name of this project.

description : string, optional

A description of the project

note : string, optional

Notes for the project

Returns **id** : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

tables : list:

```
- schema : string
- name : string
- row_count : integer
- column_count : integer
- created_at : string/time
- updated_at : string/time
```

surveys : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
```

scripts : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- type : string
- finished_at : string/time
- state : string
```

imports : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- type : string
```

```
- finished_at : string/time
- state : string
```

models : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- state : string
```

notebooks : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- current_deployment_id : integer
```

workflows : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- state : string
```

reports : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- state : string
```

script_templates : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
```

files : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- file_name : string
- file_size : integer
- expired : boolean
```

app_instances : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- slug : string
```

all_objects : list:

```
- project_id : integer
- object_id : integer
- object_type : string
- fco_type : string
- sub_type : string
- name : string
- icon : string
- author : string
- archived : string
    The archival status of the requested object(s).
```

note : string

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

put_archive (*id*, *status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **id** : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

tables : list:

```
- schema : string
- name : string
- row_count : integer
- column_count : integer
- created_at : string/time
- updated_at : string/time
```

surveys : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
```

scripts : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- type : string
- finished_at : string/time
- state : string
```

imports : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- type : string
- finished_at : string/time
- state : string
```

models : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- state : string
```

notebooks : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- current_deployment_id : integer
```

workflows : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- state : string
```

reports : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
- state : string
```

script_templates : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- name : string
```

files : list:

```
- id : integer
    The object ID.
- created_at : string/time
- updated_at : string/time
- file_name : string
- file_size : integer
- expired : boolean
```

app_instances : list:

```
- id : integer
    The object ID.
```

```
- created_at : string/time
- updated_at : string/time
- name : string
- slug : string
```

all_objects : list:

```
- project_id : integer
- object_id : integer
- object_type : string
- fco_type : string
- sub_type : string
- name : string
- icon : string
- author : string
- archived : string
    The archival status of the requested object(s).
```

note : string

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

put_shares_groups (*id*, *group_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions groups has on this object

Parameters *id* : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns *readers* : dict:

```
- users : list::
    - id : integer
    - name : string
- groups : list::
    - id : integer
    - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
 send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Queries

class `Queries` (*session_kwargs*, *return_type*='civis')

Methods

<code>delete_runs(id, run_id)</code>	Cancel a run
<code>get(id)</code>	Get details about a query
<code>get_runs(id, run_id)</code>	Check status of a run
<code>list(*[, database_id, author_id, ...])</code>	List
<code>list_runs(id, *[, limit, page_num, order, ...])</code>	List runs for the given query
<code>list_runs_logs(id, run_id, *[, last_id, limit])</code>	Get the logs for a run
<code>post(database, sql, preview_rows, *[, ...])</code>	Execute a query
<code>post_runs(id)</code>	Start a run
<code>put_scripts(id, script_id)</code>	Update the query's associated script

delete_runs (*id*, *run_id*)

Cancel a run

Parameters **id** : integer

The ID of the query.

run_id : integer

The ID of the run.

Returns None

Response code 202: success

get (*id*)

Get details about a query

Parameters *id* : integer

The query ID.

Returns *id* : integer

The query ID.

database : integer

The database ID.

sql : string

The SQL to execute.

credential : integer

The credential ID.

result_rows : list

A preview of rows returned by the query.

result_columns : list

A preview of columns returned by the query.

script_id : integer

The ID of the script associated with this query.

exception : string

Exception returned from the query, null if the query was a success.

created_at : string/time

updated_at : string/time

finished_at : string/date-time

The end time of the last run.

state : string

The state of the last run.

last_run_id : integer

The ID of the last run.

hidden : boolean

The hidden status of the object.

name : string

The name of the query.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
```

```

    This user's username.
-   initials : string
    This user's initials.
-   online : boolean
    Whether this user is online.

```

started_at : string/date-time

The start time of the last run.

report_id : integer

The ID of the report associated with this query.

get_runs (*id*, *run_id*)

Check status of a run

Parameters *id* : integer

The ID of the query.

run_id : integer

The ID of the run.

Returns *id* : integer

The ID of the run.

query_id : integer

The ID of the query.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

list (*, *database_id*='DEFAULT', *author_id*='DEFAULT', *created_before*='DEFAULT', *hidden*='DEFAULT', *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List

Parameters *database_id* : integer, optional

The database ID.

author_id : integer, optional

The author of the query.

created_before : string, optional

An upper bound for the creation date of the query.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to created_at. Must be one of: created_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The query ID.

database : integer

The database ID.

sql : string

The SQL to execute.

credential : integer

The credential ID.

result_rows : list

A preview of rows returned by the query.

result_columns : list

A preview of columns returned by the query.

script_id : integer

The ID of the script associated with this query.

exception : string

Exception returned from the query, null if the query was a success.

created_at : string/time

updated_at : string/time

finished_at : string/date-time

The end time of the last run.

state : string

The state of the last run.

last_run_id : integer

The ID of the last run.

preview_rows : integer

The number of rows to save from the query's result (maximum: 100).

started_at : string/date-time

The start time of the last run.

report_id : integer

The ID of the report associated with this query.

list_runs (*id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List runs for the given query

Parameters **id** : integer

The ID of the query.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 100.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The ID of the run.

query_id : integer

The ID of the query.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

list_runs_logs (*id, run_id, *, last_id='DEFAULT', limit='DEFAULT'*)

Get the logs for a run

Parameters id : integer

The ID of the query.

run_id : integer

The ID of the run.

last_id : integer, optional

The ID of the last log message received. Log entries with this ID value or lower will be omitted. Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.

limit : integer, optional

The maximum number of log messages to return. Default of 10000.

Returns id : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

message : string

The log message.

level : string

The level of the log. One of unknown,fatal,error,warn,info,debug.

post (*database, sql, preview_rows, *, credential='DEFAULT', hidden='DEFAULT', interactive='DEFAULT', include_header='DEFAULT', compression='DEFAULT', column_delimiter='DEFAULT', unquoted='DEFAULT', filename_prefix='DEFAULT'*)

Execute a query

Parameters database : integer

The database ID.

sql : string

The SQL to execute.

preview_rows : integer

The number of rows to save from the query's result (maximum: 100).

credential : integer, optional

The credential ID.

hidden : boolean, optional

The hidden status of the object.

interactive : boolean, optional

Deprecated and not used.

include_header : boolean, optional

Whether the CSV output should include a header row [default: true].

compression : string, optional

The type of compression. One of gzip or zip, or none [default: gzip].

column_delimiter : string, optional

The delimiter to use. One of comma or tab, or pipe [default: comma].

unquoted : boolean, optional

If true, will not quote fields.

filename_prefix : string, optional

The output filename prefix.

Returns id : integer

The query ID.

database : integer

The database ID.

sql : string

The SQL to execute.

credential : integer

The credential ID.

result_rows : list

A preview of rows returned by the query.

result_columns : list

A preview of columns returned by the query.

script_id : integer

The ID of the script associated with this query.

exception : string

Exception returned from the query, null if the query was a success.

created_at : string/time

updated_at : string/time

finished_at : string/date-time

The end time of the last run.

state : string

The state of the last run.

last_run_id : integer

The ID of the last run.

hidden : boolean

The hidden status of the object.

interactive : boolean

Deprecated and not used.

preview_rows : integer

The number of rows to save from the query's result (maximum: 100).

include_header : boolean

Whether the CSV output should include a header row [default: true].

compression : string

The type of compression. One of gzip or zip, or none [default: gzip].

column_delimiter : string

The delimiter to use. One of comma or tab, or pipe [default: comma].

unquoted : boolean

If true, will not quote fields.

filename_prefix : string

The output filename prefix.

started_at : string/date-time

The start time of the last run.

report_id : integer

The ID of the report associated with this query.

post_runs (*id*)

Start a run

Parameters id : integer

The ID of the query.

Returns id : integer

The ID of the run.

query_id : integer

The ID of the query.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

put_scripts (*id*, *script_id*)

Update the query's associated script

Parameters **id** : integer

The query ID.

script_id : integer

The ID of the script associated with this query.

Returns **id** : integer

The query ID.

database : integer

The database ID.

sql : string

The SQL to execute.

credential : integer

The credential ID.

result_rows : list

A preview of rows returned by the query.

result_columns : list

A preview of columns returned by the query.

script_id : integer

The ID of the script associated with this query.

exception : string

Exception returned from the query, null if the query was a success.

created_at : string/time

updated_at : string/time

finished_at : string/date-time

The end time of the last run.

state : string

The state of the last run.

last_run_id : integer

The ID of the last run.

hidden : boolean

The hidden status of the object.

name : string

The name of the query.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

started_at : string/date-time

The start time of the last run.

report_id : integer

The ID of the report associated with this query.

Remote_Hosts

`civis.resources._resources.Remote_Hosts`
alias of RemoteHosts

Reports

class Reports (*session_kwargs*, *return_type*='civis')

Methods

<code>delete_grants(id)</code>	Revoke permission for this report to perform Civis platform API operations on
<code>delete_projects(id, project_id)</code>	Remove a Report from a project
<code>delete_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>delete_solutions_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_solutions_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>get(id)</code>	Show a single report
<code>get_solutions(id)</code>	Show a single solutions report
<code>list(*[, type, author, template_id, hidden, ...])</code>	List
<code>list_projects(id, *[, hidden])</code>	List the projects a Report belongs to
<code>list_shares(id)</code>	List users and groups permissioned on this object
<code>list_snapshots(id)</code>	Get details about the report's snapshot automation settings
<code>list_solutions_shares(id)</code>	List users and groups permissioned on this object
<code>patch(id, *[, name, script_id, code_body, ...])</code>	Update a report
<code>patch_snapshots(id, *[, state, finished_at, ...])</code>	Update the report's snapshot automation settings
<code>post(*[, script_id, name, code_body, ...])</code>	Create a report

Continued on next page

Table 5.26 – continued from previous page

<i>post_grants</i> (id)	Grant this report the ability to perform Civis platform API operations on your
<i>post_snapshots</i> (id, *[, state, finished_at, ...])	Generate and optionally email a snapshot of the specified report
<i>put_archive</i> (id, status)	Update the archive status of this object
<i>put_projects</i> (id, project_id)	Add a Report to a project
<i>put_shares_groups</i> (id, group_ids, ...[, ...])	Set the permissions groups has on this object
<i>put_shares_users</i> (id, user_ids, ...[, ...])	Set the permissions users have on this object
<i>put_solutions_shares_groups</i> (id, group_ids, ...)	Set the permissions groups has on this object
<i>put_solutions_shares_users</i> (id, user_ids, ...)	Set the permissions users have on this object

delete_grants (*id*)

Revoke permission for this report to perform Civis platform API operations on your behalf

Parameters *id* : integer

The ID of this report.

Returns None

Response code 204: success

delete_projects (*id*, *project_id*)

Remove a Report from a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_shares_groups (*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters *id* : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users (*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters *id* : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

delete_solutions_shares_groups (*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_solutions_shares_users (*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get (*id*)

Show a single report

Parameters **id** : integer

The ID of this report.

Returns **id** : integer

The ID of this report.

name : string

The name of the report.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

projects : list:


```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

state : string

The status of the report's last run.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

The time that the report's visualization was last updated.

script : dict:

```
- id : integer
    The ID for the script.
- name : string
    The name of the script.
- sql : string
    The raw SQL query for the script.
```

job_path : string

The link to details of the job that backs this report.

tableau_id : integer

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

auth_data_url : string

auth_code_url : string

config : string

Any configuration metadata for this report.

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

api_key : string

A Civis API key that can be used by this report.

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

app_state : dict

Any application state blob for this report.

use_viewers_tableau_username : boolean

Apply user level filtering on Tableau reports.

get_solutions (*id*)

Show a single solutions report

Parameters id : integer

The ID of this report.

Returns id : integer

The ID of this report.

name : string

The name of the report.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

host : string

The host for the solution

display_url : string

The URL to display the solution report.

service_id : integer

The id of the backing service

list (*, type='DEFAULT', author='DEFAULT', template_id='DEFAULT', hidden='DEFAULT', archived='DEFAULT', limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', iterator='DEFAULT')

List

Parameters **type** : string, optional

If specified, return report of these types. It accepts a comma-separated list, possible values are 'tableau' or 'other'.

author : string, optional

If specified, return reports from this author. It accepts a comma-separated list of author ids.

template_id : integer, optional

If specified, return reports using the provided Template.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

archived : string, optional

The archival status of the requested object(s).

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to updated_at. Must be one of: updated_at, name, created_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The ID of this report.

name : string

The name of the report.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

projects : list:

A **list** of projects containing the report.

```
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

state : string

The status of the report's last run.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

The time that the report's visualization was last updated.

script : dict:

```
- id : integer
    The ID for the script.
- name : string
    The name of the script.
- sql : string
    The raw SQL query for the script.
```

job_path : string

The link to details of the job that backs this report.

tableau_id : integer

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
```

```

    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

archived : string

The archival status of the requested object(s).

list_projects (*id*, *, *hidden*='DEFAULT')

List the projects a Report belongs to

Parameters *id* : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns *id* : integer

The ID for this project.

author : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

name : string

The name of this project.

description : string

A description of the project

users : list:

```

Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

auto_share : boolean
created_at : string/time
updated_at : string/time
archived : string

The archival status of the requested object(s).

list_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_snapshots (*id*)

Get details about the report's snapshot automation settings

Parameters **id** : integer

The ID of this report.

Returns **id** : integer

The ID of this report.

state : string

The status of the job's last run.

finished_at : string/time

The time that the job's last run finished.

send_email_on_completion : boolean

Whether the job will send emails on completion.

email_template : string

Custom email template.

recipient_email_addresses : string

Email addresses to send report to, comma separated.

email_subject : string

Subject for Email.

height : integer

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

width : integer

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

parent_id : integer

The ID of the parent job that will trigger this snapshot.

list_solutions_shares (*id*)

List users and groups permissioned on this object

Parameters *id* : integer

The ID of the object.

Returns *readers* : dict:

```
- users : list::
    - id : integer
    - name : string
- groups : list::
```

```
- id : integer
- name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch (*id*, *, *name*='DEFAULT', *script_id*='DEFAULT', *code_body*='DEFAULT', *config*='DEFAULT', *app_state*='DEFAULT', *provide_api_key*='DEFAULT', *template_id*='DEFAULT', *use_viewers_tableau_username*='DEFAULT')

Update a report

Parameters *id* : integer

The ID of the report to modify.

name : string, optional

The name of the report.

script_id : integer, optional

The ID of the job (a script or a query) used to create this report.

code_body : string, optional

The code for the report visualization.

config : string, optional

app_state : dict, optional

The application state blob for this report.

provide_api_key : boolean, optional

Allow the report to provide an API key to front-end code.

template_id : integer, optional

The ID of the template used for this report. If null is passed, no template will back this report. Changes to the backing template will reset the report appState.

use_viewers_tableau_username : boolean, optional

Apply user level filtering on Tableau reports.

Returns **id** : integer

The ID of this report.

name : string

The name of the report.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

state : string

The status of the report's last run.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

The time that the report's visualization was last updated.

script : dict:

```
- id : integer
    The ID for the script.
- name : string
    The name of the script.
- sql : string
    The raw SQL query for the script.
```

job_path : string

The link to details of the job that backs this report.

tableau_id : integer

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

auth_data_url : string

auth_code_url : string

config : string

Any configuration metadata for this report.

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

api_key : string

A Civis API key that can be used by this report.

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

app_state : dict

Any application state blob for this report.

use_viewers_tableau_username : boolean

Apply user level filtering on Tableau reports.

```
patch_snapshots (id, *, state='DEFAULT', finished_at='DEFAULT',
                  send_email_on_completion='DEFAULT', email_template='DEFAULT',
                  recipient_email_addresses='DEFAULT', email_subject='DEFAULT',
                  height='DEFAULT', width='DEFAULT', schedule='DEFAULT', par-
                  ent_id='DEFAULT')
```

Update the report's snapshot automation settings

Parameters **id** : integer

The ID of this report.

state : string, optional

The status of the job's last run.

finished_at : string/time, optional

The time that the job's last run finished.

send_email_on_completion : boolean, optional

Whether the job will send emails on completion.

email_template : string, optional

Custom email template.

recipient_email_addresses : string, optional

Email addresses to send report to, comma separated.

email_subject : string, optional

Subject for Email.

height : integer, optional

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

width : integer, optional

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

parent_id : integer, optional

The ID of the parent job that will trigger this snapshot.

Returns **id** : integer

The ID of this report.

state : string

The status of the job's last run.

finished_at : string/time

The time that the job's last run finished.

send_email_on_completion : boolean

Whether the job will send emails on completion.

email_template : string

Custom email template.

recipient_email_addresses : string

Email addresses to send report to, comma separated.

email_subject : string

Subject for Email.

height : integer

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

width : integer

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

parent_id : integer

The ID of the parent job that will trigger this snapshot.

post (*, *script_id*='DEFAULT', *name*='DEFAULT', *code_body*='DEFAULT', *app_state*='DEFAULT', *provide_api_key*='DEFAULT', *template_id*='DEFAULT', *hidden*='DEFAULT')

Create a report

Parameters *script_id* : integer, optional

The ID of the job (a script or a query) used to create this report.

name : string, optional

The name of the report.

code_body : string, optional

The code for the report visualization.

app_state : dict, optional

Any application state blob for this report.

provide_api_key : boolean, optional

Allow the report to provide an API key to front-end code.

template_id : integer, optional

The ID of the template used for this report.

hidden : boolean, optional

The hidden status of the object.

Returns id : integer

The ID of this report.

name : string

The name of the report.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

state : string

The status of the report's last run.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

The time that the report's visualization was last updated.

script : dict:

```
- id : integer
    The ID for the script.
- name : string
```

```
The name of the script.  
- sql : string  
  The raw SQL query for the script.
```

job_path : string

The link to details of the job that backs this report.

tableau_id : integer

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

last_run : dict:

```
- id : integer  
- state : string  
- created_at : string/time  
  The time that the run was queued.  
- started_at : string/time  
  The time that the run started.  
- finished_at : string/time  
  The time that the run completed.  
- error : string  
  The error message for this run, if present.
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

auth_data_url : string

auth_code_url : string

config : string

Any configuration metadata for this report.

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

api_key : string

A Civis API key that can be used by this report.

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

app_state : dict

Any application state blob for this report.

use_viewers_tableau_username : boolean

Apply user level filtering on Tableau reports.

post_grants (*id*)

Grant this report the ability to perform Civis platform API operations on your behalf

Parameters **id** : integer

The ID of this report.

Returns **id** : integer

The ID of this report.

name : string

The name of the report.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

state : string

The status of the report's last run.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

The time that the report's visualization was last updated.

script : dict:

```
- id : integer
    The ID for the script.
- name : string
    The name of the script.
- sql : string
    The raw SQL query for the script.
```

job_path : string

The link to details of the job that backs this report.

tableau_id : integer

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

auth_data_url : string

auth_code_url : string

config : string

Any configuration metadata for this report.

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

api_key : string

A Civis API key that can be used by this report.

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

app_state : dict

Any application state blob for this report.

use_viewers_tableau_username : boolean

Apply user level filtering on Tableau reports.


```
post_snapshots (id, *, state='DEFAULT', finished_at='DEFAULT',
                 send_email_on_completion='DEFAULT', email_template='DEFAULT',
                 recipient_email_addresses='DEFAULT', email_subject='DEFAULT',
                 height='DEFAULT', width='DEFAULT', schedule='DEFAULT', par-
                 ent_id='DEFAULT')
```

Generate and optionally email a snapshot of the specified report

Parameters **id** : integer

The ID of this report.

state : string, optional

The status of the job's last run.

finished_at : string/time, optional

The time that the job's last run finished.

send_email_on_completion : boolean, optional

Whether the job will send emails on completion.

email_template : string, optional

Custom email template.

recipient_email_addresses : string, optional

Email addresses to send report to, comma separated.

email_subject : string, optional

Subject for Email.

height : integer, optional

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

width : integer, optional

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
    ↪ hour
```

parent_id : integer, optional

The ID of the parent job that will trigger this snapshot.

Returns **id** : integer

The ID of this report.

state : string

The status of the job's last run.

finished_at : string/time

The time that the job's last run finished.

send_email_on_completion : boolean

Whether the job will send emails on completion.

email_template : string

Custom email template.

recipient_email_addresses : string

Email addresses to send report to, comma separated.

email_subject : string

Subject for Email.

height : integer

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

width : integer

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

parent_id : integer

The ID of the parent job that will trigger this snapshot.

put_archive (*id*, *status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **id** : integer

The ID of this report.

name : string

The name of the report.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

state : string

The status of the report's last run.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

The time that the report's visualization was last updated.

script : dict:

```
- id : integer
    The ID for the script.
- name : string
    The name of the script.
- sql : string
    The raw SQL query for the script.
```

job_path : string

The link to details of the job that backs this report.

tableau_id : integer

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

auth_data_url : string

auth_code_url : string

config : string

Any configuration metadata for this report.

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

api_key : string

A Civis API key that can be used by this report.

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

app_state : dict

Any application state blob for this report.

use_viewers_tableau_username : boolean

Apply user level filtering on Tableau reports.

put_projects (*id*, *project_id*)

Add a Report to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_shares_groups (*id*, *group_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions groups has on this object

Parameters *id* : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns *readers* : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters `id` : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns `readers` : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_solutions_shares_groups (*id*, *group_ids*, *permission_level*,
*, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions groups has on this object

Parameters `id` : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_solutions_shares_users (*id*, *user_ids*, *permission_level*,
*, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns readers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Results

class Results (*session_kwargs*, *return_type*='civis')

Methods

<i>delete_grants</i> (id)	Revoke permission for this report to perform Civis platform API operations on
<i>delete_projects</i> (id, project_id)	Remove a Report from a project
<i>delete_shares_groups</i> (id, group_id)	Revoke the permissions a group has on this object
<i>delete_shares_users</i> (id, user_id)	Revoke the permissions a user has on this object
<i>delete_solutions_shares_groups</i> (id, group_id)	Revoke the permissions a group has on this object
<i>delete_solutions_shares_users</i> (id, user_id)	Revoke the permissions a user has on this object
<i>get</i> (id)	Show a single report
<i>get_solutions</i> (id)	Show a single solutions report
<i>list</i> (*[, type, author, template_id, hidden, ...])	List
<i>list_projects</i> (id, *[, hidden])	List the projects a Report belongs to
<i>list_shares</i> (id)	List users and groups permissioned on this object
<i>list_snapshots</i> (id)	Get details about the report's snapshot automation settings
<i>list_solutions_shares</i> (id)	List users and groups permissioned on this object
<i>patch</i> (id, *[, name, script_id, code_body, ...])	Update a report
<i>patch_snapshots</i> (id, *[, state, finished_at, ...])	Update the report's snapshot automation settings
<i>post</i> (*[, script_id, name, code_body, ...])	Create a report
<i>post_grants</i> (id)	Grant this report the ability to perform Civis platform API operations on your
<i>post_snapshots</i> (id, *[, state, finished_at, ...])	Generate and optionally email a snapshot of the specified report
<i>put_archive</i> (id, status)	Update the archive status of this object
<i>put_projects</i> (id, project_id)	Add a Report to a project
<i>put_shares_groups</i> (id, group_ids, ...[, ...])	Set the permissions groups has on this object
<i>put_shares_users</i> (id, user_ids, ...[, ...])	Set the permissions users have on this object
<i>put_solutions_shares_groups</i> (id, group_ids, ...)	Set the permissions groups has on this object
<i>put_solutions_shares_users</i> (id, user_ids, ...)	Set the permissions users have on this object

delete_grants (id)

Revoke permission for this report to perform Civis platform API operations on your behalf

Parameters id : integer

The ID of this report.

Returns None

Response code 204: success

delete_projects (id, project_id)

Remove a Report from a project

Parameters id : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_shares_groups (*id, group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users (*id, user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

delete_solutions_shares_groups (*id, group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_solutions_shares_users (*id, user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get (*id*)

Show a single report

Parameters **id** : integer

The ID of this report.

Returns **id** : integer

The ID of this report.

name : string

The name of the report.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

state : string

The status of the report's last run.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

The time that the report's visualization was last updated.

script : dict:

```
- id : integer
    The ID for the script.
- name : string
    The name of the script.
- sql : string
    The raw SQL query for the script.
```

job_path : string

The link to details of the job that backs this report.

tableau_id : integer

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

auth_data_url : string

auth_code_url : string

config : string

Any configuration metadata for this report.

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

api_key : string

A Civis API key that can be used by this report.

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

app_state : dict

Any application state blob for this report.

use_viewers_tableau_username : boolean

Apply user level filtering on Tableau reports.

get_solutions (*id*)

Show a single solutions report

Parameters **id** : integer

The ID of this report.

Returns **id** : integer

The ID of this report.

name : string

The name of the report.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

host : string

The host for the solution

display_url : string

The URL to display the solution report.

service_id : integer

The id of the backing service

list (*, type='DEFAULT', author='DEFAULT', template_id='DEFAULT', hidden='DEFAULT', archived='DEFAULT', limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', iterator='DEFAULT')

List

Parameters **type** : string, optional

If specified, return report of these types. It accepts a comma-separated list, possible values are 'tableau' or 'other'.

author : string, optional

If specified, return reports from this author. It accepts a comma-separated list of author ids.

template_id : integer, optional

If specified, return reports using the provided Template.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

archived : string, optional

The archival status of the requested object(s).

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to `updated_at`. Must be one of: `updated_at`, `name`, `created_at`.

order_dir : string, optional

Direction in which to sort, either `asc` (ascending) or `desc` (descending) defaulting to `desc`.

iterator : bool, optional

If `True`, return a generator to iterate over all responses. Use when more results than the maximum allowed by `limit` are needed. When `True`, `limit` and `page_num` are ignored. Defaults to `False`.

Returns **id** : integer

The ID of this report.

name : string

The name of the report.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

state : string

The status of the report's last run.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

The time that the report's visualization was last updated.

script : dict:

```
- id : integer
    The ID for the script.
- name : string
```

```

    The name of the script.
- sql : string
    The raw SQL query for the script.

```

job_path : string

The link to details of the job that backs this report.

tableau_id : integer

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

last_run : dict:

```

- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

archived : string

The archival status of the requested object(s).

list_projects (*id*, *, *hidden*='DEFAULT')

List the projects a Report belongs to

Parameters **id** : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns **id** : integer

The ID for this project.

author : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **readers** : dict:

```
- users : list::
    - id : integer
    - name : string
- groups : list::
    - id : integer
    - name : string
```

writers : dict:

```
- users : list::
    - id : integer
    - name : string
- groups : list::
    - id : integer
    - name : string
```

owners : dict:

```
- users : list::
    - id : integer
    - name : string
- groups : list::
```



```
- id : integer
- name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_snapshots (*id*)

Get details about the report's snapshot automation settings

Parameters **id** : integer

The ID of this report.

Returns **id** : integer

The ID of this report.

state : string

The status of the job's last run.

finished_at : string/time

The time that the job's last run finished.

send_email_on_completion : boolean

Whether the job will send emails on completion.

email_template : string

Custom email template.

recipient_email_addresses : string

Email addresses to send report to, comma separated.

email_subject : string

Subject for Email.

height : integer

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

width : integer

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
```

```
Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
  Alternative to scheduled minutes, number of times to run per_
↪hour
```

parent_id : integer

The ID of the parent job that will trigger this snapshot.

list_solutions_shares (*id*)

List users and groups permissioned on this object

Parameters *id* : integer

The ID of the object.

Returns *readers* : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch (*id*, *, *name*='DEFAULT', *script_id*='DEFAULT', *code_body*='DEFAULT', *config*='DEFAULT',
 app_state='DEFAULT', *provide_api_key*='DEFAULT', *template_id*='DEFAULT',
 use_viewers_tableau_username='DEFAULT')

Update a report

Parameters *id* : integer

The ID of the report to modify.

name : string, optional

The name of the report.

script_id : integer, optional

The ID of the job (a script or a query) used to create this report.

code_body : string, optional

The code for the report visualization.

config : string, optional

app_state : dict, optional

The application state blob for this report.

provide_api_key : boolean, optional

Allow the report to provide an API key to front-end code.

template_id : integer, optional

The ID of the template used for this report. If null is passed, no template will back this report. Changes to the backing template will reset the report appState.

use_viewers_tableau_username : boolean, optional

Apply user level filtering on Tableau reports.

Returns id : integer

The ID of this report.

name : string

The name of the report.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

state : string

The status of the report's last run.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

The time that the report's visualization was last updated.

script : dict:

```
- id : integer
    The ID for the script.
- name : string
    The name of the script.
- sql : string
    The raw SQL query for the script.
```

job_path : string

The link to details of the job that backs this report.

tableau_id : integer

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

auth_data_url : string

auth_code_url : string

config : string

Any configuration metadata for this report.

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

api_key : string

A Civis API key that can be used by this report.

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

app_state : dict

Any application state blob for this report.

use_viewers_tableau_username : boolean

Apply user level filtering on Tableau reports.

patch_snapshots (*id*, *, *state*=*'DEFAULT'*, *finished_at*=*'DEFAULT'*,
send_email_on_completion=*'DEFAULT'*, *email_template*=*'DEFAULT'*,
recipient_email_addresses=*'DEFAULT'*, *email_subject*=*'DEFAULT'*,
height=*'DEFAULT'*, *width*=*'DEFAULT'*, *schedule*=*'DEFAULT'*, *parent_id*=*'DEFAULT'*)

Update the report's snapshot automation settings

Parameters id : integer

The ID of this report.

state : string, optional

The status of the job's last run.

finished_at : string/time, optional

The time that the job's last run finished.

send_email_on_completion : boolean, optional

Whether the job will send emails on completion.

email_template : string, optional

Custom email template.

recipient_email_addresses : string, optional

Email addresses to send report to, comma separated.

email_subject : string, optional

Subject for Email.

height : integer, optional

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

width : integer, optional

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
```

```
Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
    ↪hour
```

parent_id : integer, optional

The ID of the parent job that will trigger this snapshot.

Returns id : integer

The ID of this report.

state : string

The status of the job's last run.

finished_at : string/time

The time that the job's last run finished.

send_email_on_completion : boolean

Whether the job will send emails on completion.

email_template : string

Custom email template.

recipient_email_addresses : string

Email addresses to send report to, comma separated.

email_subject : string

Subject for Email.

height : integer

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

width : integer

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
    ↪hour
```

parent_id : integer

The ID of the parent job that will trigger this snapshot.

post (*, *script_id='DEFAULT', name='DEFAULT', code_body='DEFAULT', app_state='DEFAULT', provide_api_key='DEFAULT', template_id='DEFAULT', hidden='DEFAULT'*)
Create a report

Parameters **script_id** : integer, optional

The ID of the job (a script or a query) used to create this report.

name : string, optional

The name of the report.

code_body : string, optional

The code for the report visualization.

app_state : dict, optional

Any application state blob for this report.

provide_api_key : boolean, optional

Allow the report to provide an API key to front-end code.

template_id : integer, optional

The ID of the template used for this report.

hidden : boolean, optional

The hidden status of the object.

Returns **id** : integer

The ID of this report.

name : string

The name of the report.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
```

```
- name : string
    The name of the project.
```

state : string

The status of the report's last run.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

The time that the report's visualization was last updated.

script : dict:

```
- id : integer
    The ID for the script.
- name : string
    The name of the script.
- sql : string
    The raw SQL query for the script.
```

job_path : string

The link to details of the job that backs this report.

tableau_id : integer

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

auth_data_url : string

auth_code_url : string

config : string

Any configuration metadata for this report.

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

api_key : string

A Civis API key that can be used by this report.

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

app_state : dict

Any application state blob for this report.

use_viewers_tableau_username : boolean

Apply user level filtering on Tableau reports.

post_grants (*id*)

Grant this report the ability to perform Civis platform API operations on your behalf

Parameters id : integer

The ID of this report.

Returns id : integer

The ID of this report.

name : string

The name of the report.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

state : string

The status of the report's last run.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

The time that the report's visualization was last updated.

script : dict:

```
- id : integer
    The ID for the script.
- name : string
    The name of the script.
- sql : string
    The raw SQL query for the script.
```

job_path : string

The link to details of the job that backs this report.

tableau_id : integer

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

auth_data_url : string

auth_code_url : string

config : string

Any configuration metadata for this report.

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

api_key : string

A Civis API key that can be used by this report.

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

app_state : dict

Any application state blob for this report.

use_viewers_tableau_username : boolean

Apply user level filtering on Tableau reports.

post_snapshots (*id*, *, *state*='DEFAULT', *finished_at*='DEFAULT',
send_email_on_completion='DEFAULT', *email_template*='DEFAULT',
recipient_email_addresses='DEFAULT', *email_subject*='DEFAULT',
height='DEFAULT', *width*='DEFAULT', *schedule*='DEFAULT', *parent_id*='DEFAULT')

Generate and optionally email a snapshot of the specified report

Parameters id : integer

The ID of this report.

state : string, optional

The status of the job's last run.

finished_at : string/time, optional

The time that the job's last run finished.

send_email_on_completion : boolean, optional

Whether the job will send emails on completion.

email_template : string, optional

Custom email template.

recipient_email_addresses : string, optional

Email addresses to send report to, comma separated.

email_subject : string, optional

Subject for Email.

height : integer, optional

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

width : integer, optional

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

parent_id : integer, optional

The ID of the parent job that will trigger this snapshot.

Returns id : integer

The ID of this report.

state : string

The status of the job's last run.

finished_at : string/time

The time that the job's last run finished.

send_email_on_completion : boolean

Whether the job will send emails on completion.

email_template : string

Custom email template.

recipient_email_addresses : string

Email addresses to send report to, comma separated.

email_subject : string

Subject for Email.

height : integer

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

width : integer

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

parent_id : integer

The ID of the parent job that will trigger this snapshot.

put_archive (*id*, *status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **id** : integer

The ID of this report.

name : string

The name of the report.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

created_at : string/time

updated_at : string/time

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

state : string

The status of the report's last run.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

The time that the report's visualization was last updated.

script : dict:

```
- id : integer
    The ID for the script.
- name : string
    The name of the script.
- sql : string
    The raw SQL query for the script.
```

job_path : string

The link to details of the job that backs this report.

tableau_id : integer

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

auth_data_url : string

auth_code_url : string

config : string

Any configuration metadata for this report.

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

api_key : string

A Civis API key that can be used by this report.

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

app_state : dict

Any application state blob for this report.

use_viewers_tableau_username : boolean

Apply user level filtering on Tableau reports.

put_projects (*id*, *project_id*)

Add a Report to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_shares_groups (*id*, *group_ids*, *permission_level*, *, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:


```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_solutions_shares_groups (*id*, *group_ids*, *permission_level*,
*, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_solutions_shares_users (*id*, *user_ids*, *permission_level*,
*, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Scripts

class Scripts (*session_kwargs*, *return_type*='civis')

Methods

<code>delete_containers_projects(id, project_id)</code>	Remove a container docker from a project
<code>delete_containers_runs(id, run_id)</code>	Cancel a run
<code>delete_containers_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_containers_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>delete_custom_projects(id, project_id)</code>	Remove a Job from a project
<code>delete_custom_runs(id, run_id)</code>	Cancel a run
<code>delete_custom_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_custom_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>delete_javascript_projects(id, project_id)</code>	Remove a scripted sql from a project
<code>delete_javascript_runs(id, run_id)</code>	Cancel a run
<code>delete_javascript_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_javascript_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>delete_python3_projects(id, project_id)</code>	Remove a python docker from a project
<code>delete_python3_runs(id, run_id)</code>	Cancel a run
<code>delete_python3_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_python3_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>delete_r_projects(id, project_id)</code>	Remove a r docker from a project
<code>delete_r_runs(id, run_id)</code>	Cancel a run
<code>delete_r_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_r_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>delete_sql_projects(id, project_id)</code>	Remove a scripts from a project
<code>delete_sql_runs(id, run_id)</code>	Cancel a run
<code>delete_sql_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object

Continued on next page

Table 5.28 – continued from previous page

<i>delete_sql_shares_users</i> (id, user_id)	Revoke the permissions a user has on this object
<i>get</i> (id)	Get details about a script
<i>get_containers</i> (id)	View a container
<i>get_containers_runs</i> (id, run_id)	Check status of a run
<i>get_custom</i> (id)	Get a Custom Script
<i>get_custom_runs</i> (id, run_id)	Check status of a run
<i>get_javascript</i> (id)	Get a JavaScript Script
<i>get_javascript_runs</i> (id, run_id)	Check status of a run
<i>get_python3</i> (id)	Get a Python Script
<i>get_python3_runs</i> (id, run_id)	Check status of a run
<i>get_r</i> (id)	Get an R Script
<i>get_r_runs</i> (id, run_id)	Check status of a run
<i>get_sql</i> (id)	Get a SQL script
<i>get_sql_runs</i> (id, run_id)	Check status of a run
<i>list</i> (*[, type, category, author, status, ...])	List scripts
<i>list_containers_projects</i> (id, *[, hidden])	List the projects a container docker belongs to
<i>list_containers_runs</i> (id, *[, limit, ...])	List runs for the given container
<i>list_containers_runs_logs</i> (id, run_id, *[, ...])	Get the logs for a run
<i>list_containers_runs_outputs</i> (id, run_id, *)	List the outputs for a run
<i>list_containers_shares</i> (id)	List users and groups permissioned on this object
<i>list_custom</i> (*[, from_template_id, author, ...])	List Custom Scripts
<i>list_custom_projects</i> (id, *[, hidden])	List the projects a Job belongs to
<i>list_custom_runs</i> (id, *[, limit, page_num, ...])	List runs for the given custom
<i>list_custom_runs_logs</i> (id, run_id, *[, ...])	Get the logs for a run
<i>list_custom_runs_outputs</i> (id, run_id, *[, ...])	List the outputs for a run
<i>list_custom_shares</i> (id)	List users and groups permissioned on this object
<i>list_history</i> (id)	Get the run history and outputs of this script
<i>list_javascript_projects</i> (id, *[, hidden])	List the projects a scripted sql belongs to
<i>list_javascript_runs</i> (id, *[, limit, ...])	List runs for the given javascript
<i>list_javascript_runs_logs</i> (id, run_id, *[, ...])	Get the logs for a run
<i>list_javascript_runs_outputs</i> (id, run_id, *)	List the outputs for a run
<i>list_javascript_shares</i> (id)	List users and groups permissioned on this object
<i>list_python3_projects</i> (id, *[, hidden])	List the projects a python docker belongs to
<i>list_python3_runs</i> (id, *[, limit, page_num, ...])	List runs for the given python
<i>list_python3_runs_logs</i> (id, run_id, *[, ...])	Get the logs for a run
<i>list_python3_runs_outputs</i> (id, run_id, *[, ...])	List the outputs for a run
<i>list_python3_shares</i> (id)	List users and groups permissioned on this object
<i>list_r_projects</i> (id, *[, hidden])	List the projects a r docker belongs to
<i>list_r_runs</i> (id, *[, limit, page_num, order, ...])	List runs for the given r
<i>list_r_runs_logs</i> (id, run_id, *[, last_id, limit])	Get the logs for a run
<i>list_r_runs_outputs</i> (id, run_id, *[, limit, ...])	List the outputs for a run
<i>list_r_shares</i> (id)	List users and groups permissioned on this object
<i>list_sql_projects</i> (id, *[, hidden])	List the projects a scripts belongs to
<i>list_sql_runs</i> (id, *[, limit, page_num, ...])	List runs for the given sql
<i>list_sql_runs_logs</i> (id, run_id, *[, last_id, ...])	Get the logs for a run

Continued on next page

Table 5.28 – continued from previous page

<i>list_sql_runs_outputs</i> (id, run_id, *[, ...])	List the outputs for a run
<i>list_sql_shares</i> (id)	List users and groups permissioned on this object
<i>list_types</i> ()	List available script types
<i>patch</i> (id, *[, name, sql, params, arguments, ...])	Update a script
<i>patch_containers</i> (id, *[, name, parent_id, ...])	Update a container
<i>patch_custom</i> (id, *[, name, parent_id, ...])	Update some attributes of this Custom Script
<i>patch_javascript</i> (id, *[, name, parent_id, ...])	Update some attributes of this JavaScript Script
<i>patch_python3</i> (id, *[, name, parent_id, ...])	Update some attributes of this Python Script
<i>patch_r</i> (id, *[, name, parent_id, ...])	Update some attributes of this R Script
<i>patch_sql</i> (id, *[, name, parent_id, ...])	Update some attributes of this SQL script
<i>post</i> (name, remote_host_id, credential_id, sql, *)	Create a script
<i>post_cancel</i> (id)	Cancel a run
<i>post_containers</i> (required_resources, ...[, ...])	Create a container
<i>post_containers_clone</i> (id, *[, ...])	Clone this Container Script
<i>post_containers_runs</i> (id)	Start a run
<i>post_containers_runs_outputs</i> (id, run_id, ...)	Add an output for a run
<i>post_custom</i> (from_template_id, *[, name, ...])	Create a Custom Script
<i>post_custom_clone</i> (id, *[, clone_schedule, ...])	Clone this Custom Script
<i>post_custom_runs</i> (id)	Start a run
<i>post_custom_runs_outputs</i> (id, run_id, ...)	Add an output for a run
<i>post_javascript</i> (name, source, ...[, ...])	Create a JavaScript Script
<i>post_javascript_clone</i> (id, *[, ...])	Clone this JavaScript Script
<i>post_javascript_runs</i> (id)	Start a run
<i>post_javascript_runs_outputs</i> (id, run_id, ...)	Add an output for a run
<i>post_python3</i> (name, source, *[, parent_id, ...])	Create a Python Script
<i>post_python3_clone</i> (id, *[, clone_schedule, ...])	Clone this Python Script
<i>post_python3_runs</i> (id)	Start a run
<i>post_python3_runs_outputs</i> (id, run_id, ...)	Add an output for a run
<i>post_r</i> (name, source, *[, parent_id, ...])	Create an R Script
<i>post_r_clone</i> (id, *[, clone_schedule, ...])	Clone this R Script
<i>post_r_runs</i> (id)	Start a run
<i>post_r_runs_outputs</i> (id, run_id, object_type, ...)	Add an output for a run
<i>post_run</i> (id)	Run a script
<i>post_sql</i> (name, sql, remote_host_id, ...[, ...])	Create a SQL script
<i>post_sql_clone</i> (id, *[, clone_schedule, ...])	Clone this SQL script
<i>post_sql_runs</i> (id)	Start a run
<i>put_containers</i> (id, required_resources, ...)	Edit a container
<i>put_containers_archive</i> (id, status)	Update the archive status of this object
<i>put_containers_projects</i> (id, project_id)	Add a container docker to a project
<i>put_containers_shares_groups</i> (id, group_ids, ...)	Set the permissions groups has on this object
<i>put_containers_shares_users</i> (id, user_ids, ...)	Set the permissions users have on this object
<i>put_custom</i> (id, *[, name, parent_id, ...])	Replace all attributes of this Custom Script
<i>put_custom_archive</i> (id, status)	Update the archive status of this object
<i>put_custom_projects</i> (id, project_id)	Add a Job to a project
<i>put_custom_shares_groups</i> (id, group_ids, ...)	Set the permissions groups has on this object

Continued on next page

Table 5.28 – continued from previous page

<code>put_custom_shares_users(id, user_ids, ...[, ...])</code>	Set the permissions users have on this object
<code>put_javascript(id, name, source, ...[, ...])</code>	Replace all attributes of this JavaScript Script
<code>put_javascript_archive(id, status)</code>	Update the archive status of this object
<code>put_javascript_projects(id, project_id)</code>	Add a scripted sql to a project
<code>put_javascript_shares_groups(id, group_ids, ...)</code>	Set the permissions groups has on this object
<code>put_javascript_shares_users(id, user_ids, ...)</code>	Set the permissions users have on this object
<code>put_python3(id, name, source, *[, ...])</code>	Replace all attributes of this Python Script
<code>put_python3_archive(id, status)</code>	Update the archive status of this object
<code>put_python3_projects(id, project_id)</code>	Add a python docker to a project
<code>put_python3_shares_groups(id, group_ids, ...)</code>	Set the permissions groups has on this object
<code>put_python3_shares_users(id, user_ids, ...)</code>	Set the permissions users have on this object
<code>put_r(id, name, source, *[, parent_id, ...])</code>	Replace all attributes of this R Script
<code>put_r_archive(id, status)</code>	Update the archive status of this object
<code>put_r_projects(id, project_id)</code>	Add a r docker to a project
<code>put_r_shares_groups(id, group_ids, ...[, ...])</code>	Set the permissions groups has on this object
<code>put_r_shares_users(id, user_ids, ...[, ...])</code>	Set the permissions users have on this object
<code>put_sql(id, name, sql, remote_host_id, ...)</code>	Replace all attributes of this SQL script
<code>put_sql_archive(id, status)</code>	Update the archive status of this object
<code>put_sql_projects(id, project_id)</code>	Add a scripts to a project
<code>put_sql_shares_groups(id, group_ids, ...[, ...])</code>	Set the permissions groups has on this object
<code>put_sql_shares_users(id, user_ids, ...[, ...])</code>	Set the permissions users have on this object

delete_containers_projects (*id, project_id*)

Remove a container docker from a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_containers_runs (*id, run_id*)

Cancel a run

Parameters *id* : integer

The ID of the container.

run_id : integer

The ID of the run.

Returns None

Response code 202: success

delete_containers_shares_groups (*id, group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_containers_shares_users (*id, user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

delete_custom_projects (*id, project_id*)

Remove a Job from a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_custom_runs (*id, run_id*)

Cancel a run

Parameters **id** : integer

The ID of the custom.

run_id : integer

The ID of the run.

Returns None

Response code 202: success

delete_custom_shares_groups (*id, group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_custom_shares_users (*id, user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

delete_javascript_projects (*id, project_id*)

Remove a scripted sql from a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_javascript_runs (*id, run_id*)

Cancel a run

Parameters **id** : integer

The ID of the javascript.

run_id : integer

The ID of the run.

Returns None

Response code 202: success

delete_javascript_shares_groups (*id, group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_javascript_shares_users (*id, user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

delete_python3_projects (*id, project_id*)

Remove a python docker from a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_python3_runs (*id, run_id*)

Cancel a run

Parameters *id* : integer

The ID of the python.

run_id : integer

The ID of the run.

Returns None

Response code 202: success

delete_python3_shares_groups (*id, group_id*)

Revoke the permissions a group has on this object

Parameters *id* : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_python3_shares_users (*id, user_id*)

Revoke the permissions a user has on this object

Parameters *id* : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

delete_r_projects (*id, project_id*)

Remove a r docker from a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_r_runs (*id*, *run_id*)

Cancel a run

Parameters **id** : integer

The ID of the r.

run_id : integer

The ID of the run.

Returns None

Response code 202: success

delete_r_shares_groups (*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_r_shares_users (*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

delete_sql_projects (*id*, *project_id*)

Remove a scripts from a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_sql_runs (*id*, *run_id*)

Cancel a run

Parameters *id* : integer

The ID of the sql.

run_id : integer

The ID of the run.

Returns None

Response code 202: success

delete_sql_shares_groups (*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters *id* : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_sql_shares_users (*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters *id* : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get (*id*)

Get details about a script

Parameters *id* : integer

The ID for the script.

Returns *id* : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time this script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
integer, float, bool, file, database, credential_aws,
credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this
↳value makes
```

```

    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.

```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```

- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.

```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour

```

notifications : dict:

```

- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.

```

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
    ↪ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
    ↪ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

sql : string

The raw SQL query for the script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

template_script_id : integer

The ID of the template script, if any.

get_containers (*id*)

View a container

Parameters **id** : integer

The ID for the script.

Returns **id** : integer

The ID for the script.

name : string

The name of the container.

type : string

The type of the script (e.g Container)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this
↳value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use
↳this
    default value. Use true, True, t, y, yes, or 1 for true bool's
↳or
    false, False, f, n, no, or 0 for false bool's. Cannot be used
↳for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

template_dependents_count : integer

How many other scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template script.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
```

```
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    ↪1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    ↪space will be used to hold the git repo configured for the
    ↪container
    ↪and anything your container writes to /tmp or /data. Fractional
    ↪values
    ↪(e.g. 0.25) are supported.
```

repo_http_uri : string

The location of a github repo to clone into the container, e.g. `github.com/my-user/my-repo.git`.

repo_ref : string

The tag or branch of the github repo to clone into the container.

remote_host_credential_id : integer

The id of the database credentials to pass into the environment of the container.

git_credential_id : integer

The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_command : string

The command to run on the container. Will be run via sh as: [`"sh"`, `"-c"`, `dockerCommand`]

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

time_zone : string

The time zone of this script.

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

target_project_id : integer

Target project to which script outputs will be added.

get_containers_runs (*id*, *run_id*)

Check status of a run

Parameters **id** : integer

The ID of the container.

run_id : integer

The ID of the run.

Returns **id** : integer

The ID of the run.

container_id : integer

The ID of the container.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

get_custom(*id*)

Get a Custom Script

Parameters *id* : integer**Returns** *id* : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g Custom)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string**projects** : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
```

```

    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used
↪for
    parameters that are required or a credential type.

```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

from_template_id : integer

The ID of the template script.

ui_report_url : integer

The url of the custom HTML.

template_script_name : string

The name of the template script.

template_note : string

The template's note.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

code_preview : string

The code that this script will run with arguments inserted.

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday

```

```
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
```

```
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

target_project_id : integer

Target project to which script outputs will be added.

get_custom_runs (*id*, *run_id*)

Check status of a run

Parameters *id* : integer

The ID of the custom.

run_id : integer

The ID of the run.

Returns *id* : integer

The ID of the run.

custom_id : integer

The ID of the custom.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

get_javascript (*id*)

Get a JavaScript Script

Parameters *id* : integer

Returns *id* : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
```



```

- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this
    ↪value makes
        this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use
    ↪this
        default value. Use true, True, t, y, yes, or 1 for true bool's
    ↪or
        false, False, f, n, no, or 0 for false bool's. Cannot be used
    ↪for
        parameters that are required or a credential type.

```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```

- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.

```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on

```

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
```

```

- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

get_javascript_runs (*id*, *run_id*)

Check status of a run

Parameters **id** : integer

The ID of the javascript.

run_id : integer

The ID of the run.

Returns **id** : integer

The ID of the run.

javascript_id : integer

The ID of the javascript.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

get_python3 (*id*)

Get a Python Script

Parameters *id* : integer**Returns** *id* : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

A definition of the parameters this script accepts **in** the arguments.

- ↪field.
- name : string
The variable's name as used within your code.
- label : string
The label to present to users when asking them **for** the value.
- description : string
A short sentence **or** fragment describing this parameter to the user.
- ↪end user.
- type : string
The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, database, credential_aws, credential_redshift, **or** credential_custom
- required : boolean
Whether this param **is** required.
- value : string
The value you would like to set this param to. Setting this value makes this parameter a fixed param.
- default : string
If an argument **for** this parameter **is not** defined, it will use this default value. Use true, **True**, t, y, yes, **or** 1 **for** true bool's.
- ↪or
false, **False**, f, n, no, **or** 0 **for** false bool's. Cannot be used for parameters that are required **or** a credential type.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

- details : string
The details link to get more information about the script.
- runs : string
The runs link to get the run information list **for** this script.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```

- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

required_resources : dict:

```

- cpu : integer
    The number of CPU shares to allocate for the container. Each_
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must_
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container._
↪This
    space will be used to hold the git repo configured for the_
↪container
    and anything your container writes to /tmp or /data. Fractional_
↪values
    (e.g. 0.25) are supported.

```

source : string

The body/text of the script.

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

get_python3_runs (*id*, *run_id*)

Check status of a run

Parameters **id** : integer

The ID of the python.

run_id : integer

The ID of the run.

Returns `id` : integer

The ID of the run.

python_id : integer

The ID of the python.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

get_r (*id*)

Get an R Script

Parameters `id` : integer

Returns `id` : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
    hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes
    successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author
    of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
```

```
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must
    ↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
```

```
space will be used to hold the git repo configured for the_  
↪container  
and anything your container writes to /tmp or /data. Fractional_  
↪values  
(e.g. 0.25) are supported.
```

source : string

The body/text of the script.

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

get_r_runs (*id*, *run_id*)

Check status of a run

Parameters **id** : integer

The ID of the r.

run_id : integer

The ID of the run.

Returns **id** : integer

The ID of the run.

r_id : integer

The ID of the r.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

get_sql (*id*)

Get a SQL script

Parameters **id** : integer

Returns **id** : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
```

```
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
```

```
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

sql : string

The raw SQL query for the script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

code_preview : string

The code that this script will run with arguments inserted.

csv_settings : dict:

```
- include_header : boolean
    Whether or not to include headers in the output data. Default: ↵
↵true
- compression : string
    The type of compression to use, if any, one of "none", "zip", or
    "gzip". Default: gzip
- column_delimiter : string
    Which delimiter to use, one of "comma", "tab", or "pipe". ↵
↵Default:
    comma
- unquoted : boolean
    Whether or not to quote fields. Default: false
- force_multifile : boolean
    Whether or not the csv should be split into multiple files. ↵
↵Default:
    false
- filename_prefix : string
    A user specified filename prefix for the output file to have. ↵
↵Default:
    null
```

get_sql_runs (*id*, *run_id*)

Check status of a run

Parameters *id* : integer

The ID of the sql.

run_id : integer

The ID of the run.

Returns id : integer

The ID of this run.

sql_id : integer

The ID of this sql.

state : string

The state of this run.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started.

finished_at : string/time

The time that this run finished.

error : string

The error message for this run, if present.

output : list:

A `list` of the outputs of this script.

- `output_name` : string
The name of the output file.
- `file_id` : integer
The unique ID of the output file.
- `path` : string
The temporary link to download this output file, valid `for 36_`
`↪hours`.

```
list(*, type='DEFAULT', category='DEFAULT', author='DEFAULT', status='DEFAULT', hidden='DEFAULT', archived='DEFAULT', limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', iterator='DEFAULT')
List scripts
```

Parameters type : string, optional

If specified, return objects of these types. The valid types are sql, python3, javascript, r, and containers.

category : string, optional

A job category for filtering scripts. Must be one of script, import, export, and enhancement.

author : string, optional

If specified, return objects from this author. Must use user IDs. A comma separated list of IDs is also accepted to return objects from multiple authors.

status : string, optional

If specified, returns objects with one of these statuses. It accepts a comma-separated list, possible values are 'running', 'failed', 'succeeded', 'idle', 'scheduled'.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

archived : string, optional

The archival status of the requested object(s).

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to updated_at. Must be one of: updated_at, name, created_at, last_run.updated_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
```

```
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

is_template : boolean

Whether others scripts use this one as a template.

from_template_id : integer

The ID of the template this script uses, if any.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

archived : string

The archival status of the requested object(s).

template_script_id : integer

The ID of the template script, if any.

list_containers_projects (*id*, *, *hidden*='DEFAULT')

List the projects a container docker belongs to

Parameters *id* : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns *id* : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_containers_runs (*id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List runs for the given container

Parameters **id** : integer

The ID of the container.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 100.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The ID of the run.

container_id : integer

The ID of the container.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

list_containers_runs_logs (*id*, *run_id*, *, *last_id*=*'DEFAULT'*, *limit*=*'DEFAULT'*)

Get the logs for a run

Parameters **id** : integer

The ID of the container.

run_id : integer

The ID of the run.

last_id : integer, optional

The ID of the last log message received. Log entries with this ID value or lower will be omitted. Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.

limit : integer, optional

The maximum number of log messages to return. Default of 10000.

Returns id : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

message : string

The log message.

level : string

The level of the log. One of unknown,fatal,error,warn,info,debug.

list_containers_runs_outputs (*id, run_id, *, limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', iterator='DEFAULT'*)

List the outputs for a run

Parameters id : integer

The ID of the container script.

run_id : integer

The ID of the run.

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to createdAt. Must be one of: createdAt, id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns object_type : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

name : string

The name of the output object.

link : string

The link to retrieve the output object.

list_containers_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_custom (*, *from_template_id*='DEFAULT', *author*='DEFAULT', *status*='DEFAULT', *hidden*='DEFAULT', *archived*='DEFAULT', *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List Custom Scripts

Parameters **from_template_id** : string, optional

If specified, return scripts based on the template with this ID. Specify multiple IDs as a comma-separated list.

author : string, optional

If specified, return objects from this author. Must use user IDs. A comma separated list of IDs is also accepted to return objects from multiple authors.

status : string, optional

If specified, returns objects with one of these statuses. It accepts a comma-separated list, possible values are 'running', 'failed', 'succeeded', 'idle', 'scheduled'.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

archived : string, optional

The archival status of the requested object(s).

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to updated_at. Must be one of: updated_at, name, created_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to asc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g Custom)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
```



```
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

from_template_id : integer

The ID of the template script.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

archived : string

The archival status of the requested object(s).

list_custom_projects (*id*, *, *hidden*=*'DEFAULT'*)

List the projects a Job belongs to

Parameters **id** : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns **id** : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_custom_runs (*id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List runs for the given custom

Parameters **id** : integer

The ID of the custom.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 100.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns id : integer

The ID of the run.

custom_id : integer

The ID of the custom.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

list_custom_runs_logs (*id*, *run_id*, *, *last_id*=*'DEFAULT'*, *limit*=*'DEFAULT'*)

Get the logs for a run

Parameters id : integer

The ID of the custom.

run_id : integer

The ID of the run.

last_id : integer, optional

The ID of the last log message received. Log entries with this ID value or lower will be omitted. Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.

limit : integer, optional

The maximum number of log messages to return. Default of 10000.

Returns id : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

message : string

The log message.

level : string

The level of the log. One of unknown,fatal,error,warn,info,debug.

list_custom_runs_outputs (*id*, *run_id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List the outputs for a run

Parameters **id** : integer

The ID of the custom script.

run_id : integer

The ID of the run.

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to created_at. Must be one of: created_at, id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **object_type** : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

name : string

The name of the output object.

link : string

The link to retrieve the output object.

list_custom_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_history (*id*)

Get the run history and outputs of this script

Parameters *id* : integer

The ID for the script.

Returns *id* : integer

The ID of this run.

sql_id : integer

The ID of this sql.

state : string

The state of this run.

is_cancel_requested : boolean

True if run cancel requested, else false.

finished_at : string/time

The time that this run finished.

error : string

The error message for this run, if present.

output : list:

```
A list of the outputs of this script.
- output_name : string
    The name of the output file.
- file_id : integer
    The unique ID of the output file.
- path : string
    The temporary link to download this output file, valid for 36
    ↪hours.
```

list_javascript_projects (*id*, *, *hidden*='DEFAULT')

List the projects a scripted sql belongs to

Parameters *id* : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns *id* : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_javascript_runs (*id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List runs for the given javascript

Parameters **id** : integer

The ID of the javascript.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 100.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The ID of the run.

javascript_id : integer

The ID of the javascript.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

list_javascript_runs_logs (*id, run_id, *, last_id='DEFAULT', limit='DEFAULT'*)

Get the logs for a run

Parameters **id** : integer

The ID of the javascript.

run_id : integer

The ID of the run.

last_id : integer, optional

The ID of the last log message received. Log entries with this ID value or lower will be omitted. Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.

limit : integer, optional

The maximum number of log messages to return. Default of 10000.

Returns **id** : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

message : string

The log message.

level : string

The level of the log. One of unknown,fatal,error,warn,info,debug.

list_javascript_runs_outputs (*id, run_id, *, limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', iterator='DEFAULT'*)

List the outputs for a run

Parameters **id** : integer

The ID of the javascript script.

run_id : integer

The ID of the run.

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to createdAt. Must be one of: createdAt, id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns `object_type` : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

name : string

The name of the output object.

link : string

The link to retrieve the output object.

list_javascript_shares (*id*)

List users and groups permissioned on this object

Parameters `id` : integer

The ID of the object.

Returns `readers` : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_python3_projects (*id*, *, *hidden*='DEFAULT')

List the projects a python docker belongs to

Parameters *id* : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns *id* : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_python3_runs (*id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List runs for the given python

Parameters **id** : integer

The ID of the python.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 100.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The ID of the run.

python_id : integer

The ID of the python.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

list_python3_runs_logs (*id*, *run_id*, *, *last_id*=*'DEFAULT'*, *limit*=*'DEFAULT'*)

Get the logs for a run

Parameters **id** : integer

The ID of the python.

run_id : integer

The ID of the run.

last_id : integer, optional

The ID of the last log message received. Log entries with this ID value or lower will be omitted. Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.

limit : integer, optional

The maximum number of log messages to return. Default of 10000.

Returns id : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

message : string

The log message.

level : string

The level of the log. One of unknown,fatal,error,warn,info,debug.

list_python3_runs_outputs (*id, run_id, *, limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', iterator='DEFAULT'*)

List the outputs for a run

Parameters id : integer

The ID of the python script.

run_id : integer

The ID of the run.

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to createdAt. Must be one of: createdAt, id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns object_type : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

name : string

The name of the output object.

link : string

The link to retrieve the output object.

list_python3_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_r_projects (*id*, *, *hidden*='DEFAULT')

List the projects a r docker belongs to

Parameters **id** : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns **id** : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_r_runs (*id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List runs for the given r

Parameters **id** : integer

The ID of the r.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 100.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns id : integer

The ID of the run.

r_id : integer

The ID of the r.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

list_r_runs_logs (*id*, *run_id*, *, *last_id*=*'DEFAULT'*, *limit*=*'DEFAULT'*)

Get the logs for a run

Parameters id : integer

The ID of the r.

run_id : integer

The ID of the run.

last_id : integer, optional

The ID of the last log message received. Log entries with this ID value or lower will be omitted. Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.

limit : integer, optional

The maximum number of log messages to return. Default of 10000.

Returns id : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

message : string

The log message.

level : string

The level of the log. One of unknown,fatal,error,warn,info,debug.

list_r_runs_outputs (*id*, *run_id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List the outputs for a run

Parameters **id** : integer

The ID of the r script.

run_id : integer

The ID of the run.

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to created_at. Must be one of: created_at, id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **object_type** : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

name : string

The name of the output object.

link : string

The link to retrieve the output object.

list_r_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **readers** : dict:


```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_sql_projects (*id*, *, *hidden*='DEFAULT')

List the projects a scripts belongs to

Parameters *id* : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns *id* : integer

The ID for this project.

author : dict:

```
- id : integer
  The ID of this user.
- name : string
  This user's name.
- username : string
  This user's username.
- initials : string
  This user's initials.
```

```
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_sql_runs (*id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List runs for the given sql

Parameters **id** : integer

The ID of the sql.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 100.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns id : integer

The ID of this run.

sql_id : integer

The ID of this sql.

state : string

The state of this run.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started.

finished_at : string/time

The time that this run finished.

error : string

The error message for this run, if present.

output : list:

```
A list of the outputs of this script.
- output_name : string
  The name of the output file.
- file_id : integer
  The unique ID of the output file.
- path : string
  The temporary link to download this output file, valid for 36
  ↪hours.
```

list_sql_runs_logs (*id*, *run_id*, *, *last_id*=*'DEFAULT'*, *limit*=*'DEFAULT'*)

Get the logs for a run

Parameters id : integer

The ID of the sql.

run_id : integer

The ID of the run.

last_id : integer, optional

The ID of the last log message received. Log entries with this ID value or lower will be omitted. Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.

limit : integer, optional

The maximum number of log messages to return. Default of 10000.

Returns id : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

message : string

The log message.

level : string

The level of the log. One of unknown,fatal,error,warn,info,debug.

list_sql_runs_outputs (*id*, *run_id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List the outputs for a run

Parameters **id** : integer

The ID of the sql script.

run_id : integer

The ID of the run.

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to created_at. Must be one of: created_at, id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **object_type** : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

name : string

The name of the output object.

link : string

The link to retrieve the output object.

list_sql_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_types ()

List available script types

Returns **name** : string

The name of the type.

patch (*id*, *, *name*='DEFAULT', *sql*='DEFAULT', *params*='DEFAULT', *arguments*='DEFAULT', *template_script_id*='DEFAULT', *schedule*='DEFAULT', *notifications*='DEFAULT', *parent_id*='DEFAULT')

Update a script

Parameters **id** : integer

The ID for the script.

name : string, optional

The name of the script.

sql : string, optional

The raw SQL query for the script.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments_  
↪field.  
Cannot be set if this script uses a template script.  
- name : string  
    The variable's name as used within your code.  
- label : string  
    The label to present to users when asking them for the value.  
- description : string  
    A short sentence or fragment describing this parameter to the_  
↪end user.  
- type : string  
    The type of parameter. Valid options: string, multi_line_string,  
    integer, float, bool, file, database, credential_aws,  
    credential_redshift, or credential_custom  
- required : boolean  
    Whether this param is required.  
- value : string  
    The value you would like to set this param to. Setting this_  
↪value makes  
    this parameter a fixed param.  
- default : string  
    If an argument for this parameter is not defined, it will use_  
↪this  
    default value. Use true, True, t, y, yes, or 1 for true bool's_  
↪or  
    false, False, f, n, no, or 0 for false bool's. Cannot be used_  
↪for  
    parameters that are required or a credential type.
```

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

template_script_id : integer, optional

The ID of the template script, if any. A script cannot both have a template script and be a template for other scripts.

schedule : dict, optional:

```
- scheduled : boolean  
    If the object is scheduled  
- scheduled_days : list  
    Day based on numeric value starting at 0 for Sunday  
- scheduled_hours : list  
    Hours of the day it is scheduled on  
- scheduled_minutes : list  
    Minutes of the day it is scheduled on  
- scheduled_runs_per_hour : integer  
    Alternative to scheduled minutes, number of times to run per_  
↪hour
```

notifications : dict, optional:

```
- urls : list  
    URLs to receive a POST request at job completion  
- success_email_subject : string  
    Custom subject line for success e-mail.
```

```

- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
    ↪ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
    ↪ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

parent_id : integer, optional

The ID of the parent job that will trigger this script

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time this script was last updated.

author : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
    hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes
    successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author
    of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

sql : string

The raw SQL query for the script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

template_script_id : integer

The ID of the template script, if any.

```
patch_containers (id, *, name='DEFAULT', parent_id='DEFAULT', user_context='DEFAULT',
                  params='DEFAULT', arguments='DEFAULT', schedule='DEFAULT',
                  notifications='DEFAULT', required_resources='DEFAULT',
                  repo_http_uri='DEFAULT', repo_ref='DEFAULT', remote_host_credential_id='DEFAULT',
                  git_credential_id='DEFAULT', docker_command='DEFAULT', docker_image_name='DEFAULT',
                  docker_image_tag='DEFAULT', cancel_timeout='DEFAULT',
                  time_zone='DEFAULT', target_project_id='DEFAULT')
```

Update a container

Parameters **id** : integer

The ID for the script.

name : string, optional

The name of the container.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.
- description : string
  A short sentence or fragment describing this parameter to the_
↪end user.
- type : string
  The type of parameter. Valid options: string, multi_line_string,
  integer, float, bool, file, database, credential_aws,
  credential_redshift, or credential_custom
- required : boolean
  Whether this param is required.
- value : string
  The value you would like to set this param to. Setting this_
↪value makes
  this parameter a fixed param.
- default : string
  If an argument for this parameter is not defined, it will use_
↪this
  default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
  false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
  parameters that are required or a credential type.
```

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳hour
```

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

required_resources : dict, optional:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each_
↳core has
    1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container._
↳This
    space will be used to hold the git repo configured for the_
↳container
    and anything your container writes to /tmp or /data. Fractional_
↳values
    (e.g. 0.25) are supported.
```

repo_http_uri : string, optional

The location of a github repo to clone into the container, e.g. `github.com/my-user/my-`

repo.git.

repo_ref : string, optional

The tag or branch of the github repo to clone into the container.

remote_host_credential_id : integer, optional

The id of the database credentials to pass into the environment of the container.

git_credential_id : integer, optional

The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_command : string, optional

The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]

docker_image_name : string, optional

The name of the docker image to pull from DockerHub.

docker_image_tag : string, optional

The tag of the docker image to pull from DockerHub (default: latest).

cancel_timeout : integer, optional

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

time_zone : string, optional

The time zone of this script.

target_project_id : integer, optional

Target project to which script outputs will be added.

Returns id : integer

The ID for the script.

name : string

The name of the container.

type : string

The type of the script (e.g Container)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
```

```
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this
↳value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use
↳this
    default value. Use true, True, t, y, yes, or 1 for true bool's
↳or
```

```
false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

template_dependents_count : integer

How many other scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template script.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
```

```
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each_
↳ core has
    1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container._
↳ This
    space will be used to hold the git repo configured for the_
↳ container
    and anything your container writes to /tmp or /data. Fractional_
↳ values
    (e.g. 0.25) are supported.
```

repo_http_uri : string

The location of a github repo to clone into the container, e.g. `github.com/my-user/my-repo.git`.

repo_ref : string

The tag or branch of the github repo to clone into the container.

remote_host_credential_id : integer

The id of the database credentials to pass into the environment of the container.

git_credential_id : integer

The id of the git credential to be used when checking out the specified git repo. If not

supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_command : string

The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

time_zone : string

The time zone of this script.

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

target_project_id : integer

Target project to which script outputs will be added.

patch_custom (*id*, *, *name*='DEFAULT', *parent_id*='DEFAULT', *arguments*='DEFAULT', *remote_host_id*='DEFAULT', *credential_id*='DEFAULT', *schedule*='DEFAULT', *notifications*='DEFAULT', *time_zone*='DEFAULT', *target_project_id*='DEFAULT')

Update some attributes of this Custom Script

Parameters **id** : integer

The ID for the script.

name : string, optional

The name of the script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

credential_id : integer, optional

The credential that this script will use.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
  job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

time_zone : string, optional

The time zone of this script.

target_project_id : integer, optional

Target project to which script outputs will be added.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g Custom)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
```

```
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used
↪for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

from_template_id : integer

The ID of the template script.

ui_report_url : integer

The url of the custom HTML.

template_script_name : string

The name of the template script.

template_note : string

The template's note.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

code_preview : string

The code that this script will run with arguments inserted.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
    successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
    of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

target_project_id : integer

Target project to which script outputs will be added.

```
patch_javascript (id, *, name='DEFAULT', parent_id='DEFAULT', user_context='DEFAULT',
                    params='DEFAULT', arguments='DEFAULT', schedule='DEFAULT', noti-
                    fications='DEFAULT', next_run_at='DEFAULT', time_zone='DEFAULT',
                    target_project_id='DEFAULT', source='DEFAULT', re-
                    mote_host_id='DEFAULT', credential_id='DEFAULT')
```

Update some attributes of this JavaScript Script

Parameters id : integer

The ID for the script.

name : string, optional

The name of the script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.
- description : string
  A short sentence or fragment describing this parameter to the_
↪end user.
- type : string
  The type of parameter. Valid options: string, multi_line_string,
  integer, float, bool, file, database, credential_aws,
  credential_redshift, or credential_custom
- required : boolean
  Whether this param is required.
- value : string
  The value you would like to set this param to. Setting this_
↪value makes
  this parameter a fixed param.
- default : string
  If an argument for this parameter is not defined, it will use_
↪this
  default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
  false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
  parameters that are required or a credential type.
```

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

next_run_at : string/time, optional

The time of the next scheduled run.

time_zone : string, optional

The time zone of this script.

target_project_id : integer, optional

Target project to which script outputs will be added.

source : string, optional

The body/text of the script.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

credential_id : integer, optional

The credential that this script will use.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:


```

A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.

```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```

- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.

```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```

- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

```

patch_python3(id, *, name='DEFAULT', parent_id='DEFAULT', user_context='DEFAULT',
               params='DEFAULT', arguments='DEFAULT', schedule='DEFAULT', notifi-
               cations='DEFAULT', next_run_at='DEFAULT', time_zone='DEFAULT', tar-
               get_project_id='DEFAULT', required_resources='DEFAULT', source='DEFAULT',
               cancel_timeout='DEFAULT')

```

Update some attributes of this Python Script

Parameters id : integer

The ID for the script.

name : string, optional

The name of the script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

```

A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string

```

```
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used
↪for
    parameters that are required or a credential type.
```

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
↪hour
```

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author
↪of the
    job.
```

```

- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

next_run_at : string/time, optional

The time of the next scheduled run.

time_zone : string, optional

The time zone of this script.

target_project_id : integer, optional

Target project to which script outputs will be added.

required_resources : dict, optional:

```

- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must
    ↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.

```

source : string, optional

The body/text of the script.

cancel_timeout : integer, optional

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments,
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
```

```

    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.

```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```

- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.

```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour

```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
↳ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
↳ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must
    ↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.
```

source : string

The body/text of the script.

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

```
patch_r(id, *, name='DEFAULT', parent_id='DEFAULT', user_context='DEFAULT',
        params='DEFAULT', arguments='DEFAULT', schedule='DEFAULT', notifi-
        cations='DEFAULT', next_run_at='DEFAULT', time_zone='DEFAULT', tar-
        get_project_id='DEFAULT', required_resources='DEFAULT', source='DEFAULT',
        cancel_timeout='DEFAULT')
```

Update some attributes of this R Script

Parameters id : integer

The ID for the script.

name : string, optional

The name of the script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments_
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the_
↳end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↳value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↳this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↳or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↳for
    parameters that are required or a credential type.
```

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳hour
```

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_from_name : string
```

```

    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

next_run_at : string/time, optional

The time of the next scheduled run.

time_zone : string, optional

The time zone of this script.

target_project_id : integer, optional

Target project to which script outputs will be added.

required_resources : dict, optional:

```

- cpu : integer
    The number of CPU shares to allocate for the container. Each_
↳ core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must_
↳ be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container._
↳ This
    space will be used to hold the git repo configured for the_
↳ container
    and anything your container writes to /tmp or /data. Fractional_
↳ values
    (e.g. 0.25) are supported.

```

source : string, optional

The body/text of the script.

cancel_timeout : integer, optional

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
```

```

- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this
    ↪value makes
        this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use
    ↪this
        default value. Use true, True, t, y, yes, or 1 for true bool's
    ↪or
        false, False, f, n, no, or 0 for false bool's. Cannot be used
    ↪for
        parameters that are required or a credential type.

```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```

- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.

```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on

```

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
```

```
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must
    ↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.
```

source : string

The body/text of the script.

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

```
patch_sql(id, *, name='DEFAULT', parent_id='DEFAULT', user_context='DEFAULT',
          params='DEFAULT', arguments='DEFAULT', schedule='DEFAULT', notifi-
          cations='DEFAULT', next_run_at='DEFAULT', time_zone='DEFAULT', tar-
          get_project_id='DEFAULT', sql='DEFAULT', remote_host_id='DEFAULT', creden-
          tial_id='DEFAULT', csv_settings='DEFAULT')
```

Update some attributes of this SQL script

Parameters **id** : integer

The ID for the script.

name : string, optional

The name of the script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.
- description : string
  A short sentence or fragment describing this parameter to the
↳end user.
- type : string
  The type of parameter. Valid options: string, multi_line_string,
integer, float, bool, file, database, credential_aws,
credential_redshift, or credential_custom
- required : boolean
  Whether this param is required.
- value : string
  The value you would like to set this param to. Setting this
↳value makes
  this parameter a fixed param.
- default : string
  If an argument for this parameter is not defined, it will use
↳this
  default value. Use true, True, t, y, yes, or 1 for true bool's
↳or
  false, False, f, n, no, or 0 for false bool's. Cannot be used
↳for
  parameters that are required or a credential type.
```

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

```
- scheduled : boolean
  If the object is scheduled
- scheduled_days : list
  Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
  Hours of the day it is scheduled on
- scheduled_minutes : list
  Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
  Alternative to scheduled minutes, number of times to run per
↳hour
```

notifications : dict, optional:

```
- urls : list
  URLs to receive a POST request at job completion
- success_email_subject : string
```



```

    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes.
    ↳ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author.
    ↳ of the
      job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

next_run_at : string/time, optional

The time of the next scheduled run.

time_zone : string, optional

The time zone of this script.

target_project_id : integer, optional

Target project to which script outputs will be added.

sql : string, optional

The raw SQL query for the script.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

credential_id : integer, optional

The credential that this script will use.

csv_settings : dict, optional:

```

- include_header : boolean
    Whether or not to include headers in the output data. Default:
    ↳ true
- compression : string
    The type of compression to use, if any, one of "none", "zip", or
    "gzip". Default: gzip
- column_delimiter : string
    Which delimiter to use, one of "comma", "tab", or "pipe".
    ↳ Default:
      comma
- unquoted : boolean
    Whether or not to quote fields. Default: false
- force_multifile : boolean
    Whether or not the csv should be split into multiple files.
    ↳ Default:

```

```
    false
- filename_prefix : string
    A user specified filename prefix for the output file to have.
↪Default:
    null
```

Returns `id` : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.
- description : string
  A short sentence or fragment describing this parameter to the_
↪end user.
- type : string
  The type of parameter. Valid options: string, multi_line_string,
integer, float, bool, file, database, credential_aws,
credential_redshift, or credential_custom
- required : boolean
  Whether this param is required.
- value : string
  The value you would like to set this param to. Setting this_
↪value makes
  this parameter a fixed param.
- default : string
  If an argument for this parameter is not defined, it will use_
↪this
  default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
  false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
  parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
  The details link to get more information about the script.
- runs : string
  The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

sql : string

The raw SQL query for the script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

code_preview : string

The code that this script will run with arguments inserted.

csv_settings : dict:

```
- include_header : boolean
  Whether or not to include headers in the output data. Default: ↵
↵true
- compression : string
  The type of compression to use, if any, one of "none", "zip", or
  "gzip". Default: gzip
- column_delimiter : string
  Which delimiter to use, one of "comma", "tab", or "pipe". ↵
↵Default:
  comma
- unquoted : boolean
  Whether or not to quote fields. Default: false
- force_multifile : boolean
  Whether or not the csv should be split into multiple files. ↵
↵Default:
  false
- filename_prefix : string
```

```
A user specified filename prefix for the output file to have.↵
↵Default:
    null
```

post (*name*, *remote_host_id*, *credential_id*, *sql*, *, *params*=*'DEFAULT'*, *arguments*=*'DEFAULT'*, *template_script_id*=*'DEFAULT'*, *notifications*=*'DEFAULT'*, *hidden*=*'DEFAULT'*)
Create a script

Parameters *name* : string

The name of the script.

remote_host_id : integer

The database ID.

credential_id : integer

The credential ID.

sql : string

The raw SQL query for the script.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments.↵
↵field.
Cannot be set if this script uses a template script.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the↵
↵end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this↵
↵value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use↵
↵this
    default value. Use true, True, t, y, yes, or 1 for true bool's↵
↵or
    false, False, f, n, no, or 0 for false bool's. Cannot be used↵
↵for
    parameters that are required or a credential type.
```

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

template_script_id : integer, optional

The ID of the template script, if any. A script cannot both have a template script and be a template for other scripts.

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳of the
  job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

hidden : boolean, optional

The hidden status of the object.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
```

```
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this
↳value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use
↳this
    default value. Use true, True, t, y, yes, or 1 for true bool's
↳or
    false, False, f, n, no, or 0 for false bool's. Cannot be used
↳for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
    hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes
    successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author
    of the
    job.
- failure_email_addresses : list
```

```
Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

template_script_id : integer

The ID of the template script, if any.

post_cancel (*id*)

Cancel a run

Parameters **id** : integer

The ID of the job.

Returns `id` : integer

The ID of the run.

state : string

The state of the run, one of 'queued', 'running' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

post_containers (*required_resources*, *docker_command*, *docker_image_name*, *,
name='DEFAULT', *parent_id*='DEFAULT', *user_context*='DEFAULT',
params='DEFAULT', *arguments*='DEFAULT', *schedule*='DEFAULT', *no-*
tifications='DEFAULT', *repo_http_uri*='DEFAULT', *repo_ref*='DEFAULT',
remote_host_credential_id='DEFAULT', *git_credential_id*='DEFAULT',
docker_image_tag='DEFAULT', *cancel_timeout*='DEFAULT',
time_zone='DEFAULT', *hidden*='DEFAULT', *target_project_id*='DEFAULT')

Create a container

Parameters `required_resources` : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.
```

docker_command : string

The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]

docker_image_name : string

The name of the docker image to pull from DockerHub.

name : string, optional

The name of the container.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

"runner" or "author", who to execute the script as when run as a template.

params : list, optional:

A definition of the parameters this script accepts **in** the arguments.
↪field.

- name : string
The variable's name as used within your code.
- label : string
The label to present to users when asking them **for** the value.
- description : string
A short sentence **or** fragment describing this parameter to the
↪end user.
- type : string
The **type** of parameter. Valid options: string, multi_line_string,
integer, float, bool, file, database, credential_aws,
credential_redshift, **or** credential_custom
- required : boolean
Whether this param **is** required.
- value : string
The value you would like to **set** this param to. Setting this
↪value makes
this parameter a fixed param.
- default : string
If an argument **for** this parameter **is not** defined, it will use
↪this
default value. Use true, **True**, t, y, yes, **or** 1 **for** true bool's.
↪or
false, **False**, f, n, no, **or** 0 **for** false bool's. Cannot be used
↪for
parameters that are required **or** a credential type.

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

- scheduled : boolean
If the **object is** scheduled
- scheduled_days : list
Day based on numeric value starting at 0 **for** Sunday
- scheduled_hours : list
Hours of the day it **is** scheduled on
- scheduled_minutes : list
Minutes of the day it **is** scheduled on
- scheduled_runs_per_hour : integer
Alternative to scheduled minutes, number of times to run per
↪hour

notifications : dict, optional:

- urls : list
URLs to receive a POST request at job completion
- success_email_subject : string
Custom subject line **for** success e-mail.
- success_email_body : string
Custom body text **for** success e-mail, written **in** Markdown.
- success_email_addresses : list
Addresses to notify by e-mail when the job completes
↪successfully.
- success_email_from_name : string

```

    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

repo_http_uri : string, optional

The location of a github repo to clone into the container, e.g. github.com/my-user/my-repo.git.

repo_ref : string, optional

The tag or branch of the github repo to clone into the container.

remote_host_credential_id : integer, optional

The id of the database credentials to pass into the environment of the container.

git_credential_id : integer, optional

The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_image_tag : string, optional

The tag of the docker image to pull from DockerHub (default: latest).

cancel_timeout : integer, optional

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

time_zone : string, optional

The time zone of this script.

hidden : boolean, optional

The hidden status of the object.

target_project_id : integer, optional

Target project to which script outputs will be added.

Returns id : integer

The ID for the script.

name : string

The name of the container.

type : string

The type of the script (e.g Container)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
integer, float, bool, file, database, credential_aws,
credential_redshift, or credential_custom
```

```

- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.

```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

template_dependents_count : integer

How many other scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template script.

template_script_name : string

The name of the template script.

links : dict:

```

- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.

```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour

```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
↳ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
↳ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each,
↳ core has
    1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
↳ This
    space will be used to hold the git repo configured for the,
↳ container
    and anything your container writes to /tmp or /data. Fractional,
↳ values
    (e.g. 0.25) are supported.
```

repo_http_uri : string

The location of a github repo to clone into the container, e.g. `github.com/my-user/my-repo.git`.

repo_ref : string

The tag or branch of the github repo to clone into the container.

remote_host_credential_id : integer

The id of the database credentials to pass into the environment of the container.

git_credential_id : integer

The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_command : string

The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

time_zone : string

The time zone of this script.

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

target_project_id : integer

Target project to which script outputs will be added.

post_containers_clone (*id*, *, *clone_schedule*='DEFAULT', *clone_triggers*='DEFAULT',
clone_notifications='DEFAULT')

Clone this Container Script

Parameters id : integer

The ID for the script.

clone_schedule : boolean, optional

If true, also copy the schedule to the new script.

clone_triggers : boolean, optional

If true, also copy the triggers to the new script.

clone_notifications : boolean, optional

If true, also copy the notifications to the new script.

Returns **id** : integer

The ID for the script.

name : string

The name of the container.

type : string

The type of the script (e.g Container)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.
- description : string
  A short sentence or fragment describing this parameter to the
↳end user.
- type : string
  The type of parameter. Valid options: string, multi_line_string,
  integer, float, bool, file, database, credential_aws,
  credential_redshift, or credential_custom
- required : boolean
  Whether this param is required.
- value : string
  The value you would like to set this param to. Setting this
↳value makes
  this parameter a fixed param.
- default : string
  If an argument for this parameter is not defined, it will use
↳this
  default value. Use true, True, t, y, yes, or 1 for true bool's
↳or
  false, False, f, n, no, or 0 for false bool's. Cannot be used
↳for
  parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

template_dependents_count : integer

How many other scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template script.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.
```

repo_http_uri : string

The location of a github repo to clone into the container, e.g. github.com/my-user/my-repo.git.

repo_ref : string

The tag or branch of the github repo to clone into the container.

remote_host_credential_id : integer

The id of the database credentials to pass into the environment of the container.

git_credential_id : integer

The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_command : string

The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
```

```
The time that the run completed.
- error : string
  The error message for this run, if present.
```

time_zone : string

The time zone of this script.

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

target_project_id : integer

Target project to which script outputs will be added.

post_containers_runs (*id*)

Start a run

Parameters id : integer

The ID of the container.

Returns id : integer

The ID of the run.

container_id : integer

The ID of the container.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

post_containers_runs_outputs (*id, run_id, object_type, object_id*)

Add an output for a run

Parameters id : integer

The ID of the container script.

run_id : integer

The ID of the run.

object_type : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

Returns object_type : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

name : string

The name of the output object.

link : string

The link to retrieve the output object.

post_custom (*from_template_id*, *, *name*='DEFAULT', *parent_id*='DEFAULT', *arguments*='DEFAULT', *remote_host_id*='DEFAULT', *credential_id*='DEFAULT', *schedule*='DEFAULT', *notifications*='DEFAULT', *time_zone*='DEFAULT', *hidden*='DEFAULT', *target_project_id*='DEFAULT')

Create a Custom Script

Parameters from_template_id : integer

The ID of the template script.

name : string, optional

The name of the script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

credential_id : integer, optional

The credential that this script will use.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
    hour
```

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
↳ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
↳ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

time_zone : string, optional

The time zone of this script.

hidden : boolean, optional

The hidden status of the object.

target_project_id : integer, optional

Target project to which script outputs will be added.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g Custom)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
```



```
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

projects : list:

A `list` of projects containing the script.

```
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

params : list:

A definition of the parameters this script accepts `in` the arguments, `↵`field.

```
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the ↵end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, database, credential_aws, credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this ↵value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use ↵this
    default value. Use true, True, t, y, yes, or 1 for true bool's ↵or
    false, False, f, n, no, or 0 for false bool's. Cannot be used ↵for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

from_template_id : integer

The ID of the template script.

ui_report_url : integer

The url of the custom HTML.

template_script_name : string

The name of the template script.

template_note : string

The template's note.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

code_preview : string

The code that this script will run with arguments inserted.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
```

```

    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

running_as : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

time_zone : string

The time zone of this script.

last_run : dict:

```

- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

target_project_id : integer

Target project to which script outputs will be added.

```

post_custom_clone (id, *, clone_schedule='DEFAULT', clone_triggers='DEFAULT',
                    clone_notifications='DEFAULT')
Clone this Custom Script

```

Parameters **id** : integer

The ID for the script.

clone_schedule : boolean, optional

If true, also copy the schedule to the new script.

clone_triggers : boolean, optional

If true, also copy the triggers to the new script.

clone_notifications : boolean, optional

If true, also copy the notifications to the new script.

Returns `id` : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g Custom)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

params : list:

```
A definition of the parameters this script accepts in the arguments.
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
```

```

- description : string
    A short sentence or fragment describing this parameter to the
    ↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this
    ↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use
    ↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's
    ↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used
    ↪for
    parameters that are required or a credential type.

```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

from_template_id : integer

The ID of the template script.

ui_report_url : integer

The url of the custom HTML.

template_script_name : string

The name of the template script.

template_note : string

The template's note.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

code_preview : string

The code that this script will run with arguments inserted.

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list

```

```
Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
  Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
  URLs to receive a POST request at job completion
- success_email_subject : string
  Custom subject line for success e-mail.
- success_email_body : string
  Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
  Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
  Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
  Address for replies to success emails; defaults to the author_
↪of the
  job.
- failure_email_addresses : list
  Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
  Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
  If success email notifications are on.
- failure_on : boolean
  If failure email notifications are on.
```

running_as : dict:

```
- id : integer
  The ID of this user.
- name : string
  This user's name.
- username : string
  This user's username.
- initials : string
  This user's initials.
- online : boolean
  Whether this user is online.
```

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
```

```
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

target_project_id : integer

Target project to which script outputs will be added.

post_custom_runs (*id*)

Start a run

Parameters id : integer

The ID of the custom.

Returns id : integer

The ID of the run.

custom_id : integer

The ID of the custom.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

post_custom_runs_outputs (*id, run_id, object_type, object_id*)

Add an output for a run

Parameters id : integer

The ID of the custom script.

run_id : integer

The ID of the run.

object_type : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

Returns `object_type` : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

name : string

The name of the output object.

link : string

The link to retrieve the output object.

```
post_javascript (name, source, remote_host_id, credential_id, *, parent_id='DEFAULT',
                  user_context='DEFAULT', params='DEFAULT', arguments='DEFAULT',
                  schedule='DEFAULT', notifications='DEFAULT', next_run_at='DEFAULT',
                  time_zone='DEFAULT', hidden='DEFAULT', target_project_id='DEFAULT')
```

Create a JavaScript Script

Parameters `name` : string

The name of the script.

source : string

The body/text of the script.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.
- description : string
  A short sentence or fragment describing this parameter to the_
↪end user.
- type : string
  The type of parameter. Valid options: string, multi_line_string,
  integer, float, bool, file, database, credential_aws,
  credential_redshift, or credential_custom
- required : boolean
  Whether this param is required.
- value : string
  The value you would like to set this param to. Setting this_
↪value makes
```



```

    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's.
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used
↪for
    parameters that are required or a credential type.

```

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
↪hour

```

notifications : dict, optional:

```

- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

next_run_at : string/time, optional

The time of the next scheduled run.

time_zone : string, optional

The time zone of this script.

hidden : boolean, optional

The hidden status of the object.

target_project_id : integer, optional

Target project to which script outputs will be added.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.
- description : string
  A short sentence or fragment describing this parameter to the
↳end user.
- type : string
  The type of parameter. Valid options: string, multi_line_string,
integer, float, bool, file, database, credential_aws,
credential_redshift, or credential_custom
- required : boolean
  Whether this param is required.
- value : string
  The value you would like to set this param to. Setting this
↳value makes
  this parameter a fixed param.
- default : string
  If an argument for this parameter is not defined, it will use
↳this
  default value. Use true, True, t, y, yes, or 1 for true bool's
↳or
  false, False, f, n, no, or 0 for false bool's. Cannot be used
↳for
  parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

post_javascript_clone (*id*, *, *clone_schedule*='DEFAULT', *clone_triggers*='DEFAULT',
clone_notifications='DEFAULT')

Clone this JavaScript Script

Parameters **id** : integer

The ID for the script.

clone_schedule : boolean, optional

If true, also copy the schedule to the new script.

clone_triggers : boolean, optional

If true, also copy the triggers to the new script.

clone_notifications : boolean, optional

If true, also copy the notifications to the new script.

Returns **id** : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
```

```

- description : string
    A short sentence or fragment describing this parameter to the
    ↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this
    ↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use
    ↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's
    ↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used
    ↪for
    parameters that are required or a credential type.

```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```

- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.

```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on

```

```
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
```



```

- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

post_javascript_runs (*id*)

Start a run

Parameters id : integer

The ID of the javascript.

Returns id : integer

The ID of the run.

javascript_id : integer

The ID of the javascript.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

post_javascript_runs_outputs (*id, run_id, object_type, object_id*)

Add an output for a run

Parameters `id` : integer

The ID of the javascript script.

run_id : integer

The ID of the run.

object_type : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

Returns `object_type` : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

name : string

The name of the output object.

link : string

The link to retrieve the output object.

```
post_python3 (name, source, *, parent_id='DEFAULT', user_context='DEFAULT',
               params='DEFAULT', arguments='DEFAULT', schedule='DEFAULT', notifications='DEFAULT',
               next_run_at='DEFAULT', time_zone='DEFAULT', hidden='DEFAULT', target_project_id='DEFAULT',
               required_resources='DEFAULT', cancel_timeout='DEFAULT')
```

Create a Python Script

Parameters `name` : string

The name of the script.

source : string

The body/text of the script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments_
↳field.
- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.
- description : string
  A short sentence or fragment describing this parameter to the
↳end user.
- type : string
  The type of parameter. Valid options: string, multi_line_string,
```

```

integer, float, bool, file, database, credential_aws,
credential_redshift, or credential_custom
- required : boolean
  Whether this param is required.
- value : string
  The value you would like to set this param to. Setting this
↪value makes
  this parameter a fixed param.
- default : string
  If an argument for this parameter is not defined, it will use
↪this
  default value. Use true, True, t, y, yes, or 1 for true bool's
↪or
  false, False, f, n, no, or 0 for false bool's. Cannot be used
↪for
  parameters that are required or a credential type.

```

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

```

- scheduled : boolean
  If the object is scheduled
- scheduled_days : list
  Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
  Hours of the day it is scheduled on
- scheduled_minutes : list
  Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
  Alternative to scheduled minutes, number of times to run per
↪hour

```

notifications : dict, optional:

```

- urls : list
  URLs to receive a POST request at job completion
- success_email_subject : string
  Custom subject line for success e-mail.
- success_email_body : string
  Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
  Addresses to notify by e-mail when the job completes
↪successfully.
- success_email_from_name : string
  Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
  Address for replies to success emails; defaults to the author
↪of the
  job.
- failure_email_addresses : list
  Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
  Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
  If success email notifications are on.

```

```
- failure_on : boolean
    If failure email notifications are on.
```

next_run_at : string/time, optional

The time of the next scheduled run.

time_zone : string, optional

The time zone of this script.

hidden : boolean, optional

The hidden status of the object.

target_project_id : integer, optional

Target project to which script outputs will be added.

required_resources : dict, optional:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must
    ↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.
```

cancel_timeout : integer, optional

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

A **list** of projects containing the script.

```

- id : integer
    The ID for the project.
- name : string
    The name of the project.

```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

A definition of the parameters this script accepts **in** the arguments, **↵**field.

```

- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the ↵end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, database, credential_aws, credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this ↵value makes this parameter a fixed param.
- default : string

```

```
If an argument for this parameter is not defined, it will use_  
↪this  
default value. Use true, True, t, y, yes, or 1 for true bool's_  
↪or  
false, False, f, n, no, or 0 for false bool's. Cannot be used_  
↪for  
parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string  
    The details link to get more information about the script.  
- runs : string  
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean  
    If the object is scheduled  
- scheduled_days : list  
    Day based on numeric value starting at 0 for Sunday  
- scheduled_hours : list  
    Hours of the day it is scheduled on  
- scheduled_minutes : list  
    Minutes of the day it is scheduled on  
- scheduled_runs_per_hour : integer  
    Alternative to scheduled minutes, number of times to run per_  
↪hour
```

notifications : dict:

```
- urls : list  
    URLs to receive a POST request at job completion  
- success_email_subject : string  
    Custom subject line for success e-mail.  
- success_email_body : string  
    Custom body text for success e-mail, written in Markdown.
```

```

- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
    successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
    of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

running_as : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```

- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must
    ↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.
```

source : string

The body/text of the script.

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

post_python3_clone (*id*, *, *clone_schedule*='DEFAULT', *clone_triggers*='DEFAULT',
 clone_notifications='DEFAULT')

Clone this Python Script

Parameters *id* : integer

The ID for the script.

clone_schedule : boolean, optional

If true, also copy the schedule to the new script.

clone_triggers : boolean, optional

If true, also copy the triggers to the new script.

clone_notifications : boolean, optional

If true, also copy the notifications to the new script.

Returns *id* : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
```

```
The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
```

```

- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
    ↪ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
    ↪ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

running_as : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```

- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must
    ↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.
```

source : string

The body/text of the script.

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

post_python3_runs (*id*)

Start a run

Parameters **id** : integer

The ID of the python.

Returns **id** : integer

The ID of the run.

python_id : integer

The ID of the python.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

post_python3_runs_outputs (*id, run_id, object_type, object_id*)

Add an output for a run

Parameters **id** : integer

The ID of the python script.

run_id : integer

The ID of the run.

object_type : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

Returns **object_type** : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

name : string

The name of the output object.

link : string

The link to retrieve the output object.

post_r (*name, source, *, parent_id='DEFAULT', user_context='DEFAULT', params='DEFAULT', arguments='DEFAULT', schedule='DEFAULT', notifications='DEFAULT', next_run_at='DEFAULT', time_zone='DEFAULT', hidden='DEFAULT', target_project_id='DEFAULT', required_resources='DEFAULT', cancel_timeout='DEFAULT'*)

Create an R Script

Parameters **name** : string

The name of the script.

source : string

The body/text of the script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
  The variable's name as used within your code.
```

```
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used
↪for
    parameters that are required or a credential type.
```

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
↪hour
```

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author
↪of the
```

```

    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

next_run_at : string/time, optional

The time of the next scheduled run.

time_zone : string, optional

The time zone of this script.

hidden : boolean, optional

The hidden status of the object.

target_project_id : integer, optional

Target project to which script outputs will be added.

required_resources : dict, optional:

```

- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must
    ↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.

```

cancel_timeout : integer, optional

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments,
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
```



```

    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.

```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```

- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.

```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour

```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
↳ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
↳ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must
    ↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.
```

source : string

The body/text of the script.

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

```
post_r_clone (id, *, clone_schedule='DEFAULT', clone_triggers='DEFAULT',
               clone_notifications='DEFAULT')
Clone this R Script
```

Parameters **id** : integer

The ID for the script.

clone_schedule : boolean, optional

If true, also copy the schedule to the new script.

clone_triggers : boolean, optional

If true, also copy the triggers to the new script.

clone_notifications : boolean, optional

If true, also copy the notifications to the new script.

Returns **id** : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
```

```

- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.

```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```

- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.

```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on

```

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
```

```
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must
    ↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.
```

source : string

The body/text of the script.

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

post_r_runs (*id*)

Start a run

Parameters **id** : integer

The ID of the r.

Returns **id** : integer

The ID of the run.

r_id : integer

The ID of the r.

state : string

The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started at.

finished_at : string/time

The time the last run completed.

error : string

The error, if any, returned by the run.

post_r_runs_outputs (*id, run_id, object_type, object_id*)

Add an output for a run

Parameters **id** : integer

The ID of the r script.

run_id : integer

The ID of the run.

object_type : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

Returns **object_type** : string

The type of the output. Valid values are File, Table, Report, Project, or Credential

object_id : integer

The ID of the output object.

name : string

The name of the output object.

link : string

The link to retrieve the output object.

post_run (*id*)

Run a script

Parameters **id** : integer

The ID for the script.

Returns None

Response code 204: success

post_sql (*name, sql, remote_host_id, credential_id, *, parent_id='DEFAULT',
user_context='DEFAULT', params='DEFAULT', arguments='DEFAULT',
schedule='DEFAULT', notifications='DEFAULT', next_run_at='DEFAULT',
time_zone='DEFAULT', hidden='DEFAULT', target_project_id='DEFAULT',
csv_settings='DEFAULT'*)

Create a SQL script

Parameters **name** : string

The name of the script.

sql : string

The raw SQL query for the script.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.
- description : string
  A short sentence or fragment describing this parameter to the
↳end user.
- type : string
  The type of parameter. Valid options: string, multi_line_string,
  integer, float, bool, file, database, credential_aws,
  credential_redshift, or credential_custom
- required : boolean
  Whether this param is required.
- value : string
  The value you would like to set this param to. Setting this
↳value makes
  this parameter a fixed param.
- default : string
  If an argument for this parameter is not defined, it will use
↳this
  default value. Use true, True, t, y, yes, or 1 for true bool's
↳or
  false, False, f, n, no, or 0 for false bool's. Cannot be used
↳for
  parameters that are required or a credential type.
```

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

```
- scheduled : boolean
  If the object is scheduled
- scheduled_days : list
  Day based on numeric value starting at 0 for Sunday
```

```
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳hour
```

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

next_run_at : string/time, optional

The time of the next scheduled run.

time_zone : string, optional

The time zone of this script.

hidden : boolean, optional

The hidden status of the object.

target_project_id : integer, optional

Target project to which script outputs will be added.

csv_settings : dict, optional:

```
- include_header : boolean
    Whether or not to include headers in the output data. Default:_
↳true
- compression : string
    The type of compression to use, if any, one of "none", "zip", or
    "gzip". Default: gzip
- column_delimiter : string
    Which delimiter to use, one of "comma", "tab", or "pipe"._
↳Default:
```

```

    comma
-   unquoted : boolean
    Whether or not to quote fields. Default: false
-   force_multifile : boolean
    Whether or not the csv should be split into multiple files.
↪Default:
    false
-   filename_prefix : string
    A user specified filename prefix for the output file to have.
↪Default:
    null

```

Returns `id` : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```

-   id : integer
    The ID of this user.
-   name : string
    This user's name.
-   username : string
    This user's username.
-   initials : string
    This user's initials.
-   online : boolean
    Whether this user is online.

```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```

A list of projects containing the script.
-   id : integer
    The ID for the project.
-   name : string
    The name of the project.

```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments.  
↪field.  
- name : string  
  The variable's name as used within your code.  
- label : string  
  The label to present to users when asking them for the value.  
- description : string  
  A short sentence or fragment describing this parameter to the  
↪end user.  
- type : string  
  The type of parameter. Valid options: string, multi_line_string,  
integer, float, bool, file, database, credential_aws,  
credential_redshift, or credential_custom  
- required : boolean  
  Whether this param is required.  
- value : string  
  The value you would like to set this param to. Setting this  
↪value makes  
  this parameter a fixed param.  
- default : string  
  If an argument for this parameter is not defined, it will use  
↪this  
  default value. Use true, True, t, y, yes, or 1 for true bool's  
↪or  
  false, False, f, n, no, or 0 for false bool's. Cannot be used  
↪for  
  parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

- details : string
The details link to get more information about the script.
- runs : string
The runs link to get the run information **list** **for** this script.

schedule : dict:

- scheduled : boolean
If the **object** **is** scheduled
- scheduled_days : **list**
Day based on numeric value starting at 0 **for** Sunday
- scheduled_hours : **list**
Hours of the day it **is** scheduled on
- scheduled_minutes : **list**
Minutes of the day it **is** scheduled on
- scheduled_runs_per_hour : integer
Alternative to scheduled minutes, number of times to run per_
↪hour

notifications : dict:

- urls : **list**
URLs to receive a POST request at job completion
- success_email_subject : string
Custom subject line **for** success e-mail.
- success_email_body : string
Custom body text **for** success e-mail, written **in** Markdown.
- success_email_addresses : **list**
Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
Name **from which** success emails are sent; defaults to "Civis."
- success_email_reply_to : string
Address **for** replies to success emails; defaults to the author_
↪of the
job.
- failure_email_addresses : **list**
Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
If success email notifications are on.
- failure_on : boolean
If failure email notifications are on.

running_as : dict:

- id : integer
The ID of this user.
- name : string
This user's **name**.
- username : string
This user's **username**.
- initials : string
This user's **initials**.
- online : boolean
Whether this user **is** online.

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

sql : string

The raw SQL query for the script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

code_preview : string

The code that this script will run with arguments inserted.

csv_settings : dict:

```
- include_header : boolean
  Whether or not to include headers in the output data. Default: ↵
↵true
- compression : string
  The type of compression to use, if any, one of "none", "zip", or
  "gzip". Default: gzip
- column_delimiter : string
  Which delimiter to use, one of "comma", "tab", or "pipe". ↵
↵Default:
```

```

    comma
-   unquoted : boolean
    Whether or not to quote fields. Default: false
-   force_multifile : boolean
    Whether or not the csv should be split into multiple files.
↪Default:
    false
-   filename_prefix : string
    A user specified filename prefix for the output file to have.
↪Default:
    null

```

```

post_sql_clone(id, *, clone_schedule='DEFAULT', clone_triggers='DEFAULT',
               clone_notifications='DEFAULT')

```

Clone this SQL script

Parameters **id** : integer

The ID for the script.

clone_schedule : boolean, optional

If true, also copy the schedule to the new script.

clone_triggers : boolean, optional

If true, also copy the triggers to the new script.

clone_notifications : boolean, optional

If true, also copy the notifications to the new script.

Returns **id** : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```

-   id : integer
    The ID of this user.
-   name : string
    This user's name.
-   username : string
    This user's username.
-   initials : string
    This user's initials.
-   online : boolean
    Whether this user is online.

```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↳value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↳this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↳or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↳for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
    hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes
    successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author
    of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
```

```
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

sql : string

The raw SQL query for the script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

code_preview : string

The code that this script will run with arguments inserted.

csv_settings : dict:

```
- include_header : boolean
    Whether or not to include headers in the output data. Default: ↵
↵true
- compression : string
    The type of compression to use, if any, one of "none", "zip", or
    "gzip". Default: gzip
- column_delimiter : string
    Which delimiter to use, one of "comma", "tab", or "pipe". ↵
↵Default:
    comma
- unquoted : boolean
    Whether or not to quote fields. Default: false
- force_multifile : boolean
    Whether or not the csv should be split into multiple files. ↵
↵Default:
    false
- filename_prefix : string
    A user specified filename prefix for the output file to have. ↵
↵Default:
    null
```

post_sql_runs (*id*)

Start a run

Parameters **id** : integer

The ID of the sql.

Returns **id** : integer

The ID of this run.

sql_id : integer

The ID of this sql.

state : string

The state of this run.

is_cancel_requested : boolean

True if run cancel requested, else false.

started_at : string/time

The time the last run started.

finished_at : string/time

The time that this run finished.

error : string

The error message for this run, if present.

output : list:

```
A list of the outputs of this script.
- output_name : string
    The name of the output file.
- file_id : integer
    The unique ID of the output file.
- path : string
    The temporary link to download this output file, valid for 36
    ↪hours.
```

```
put_containers(id, required_resources, docker_command, docker_image_name, *,
               name='DEFAULT', parent_id='DEFAULT', user_context='DEFAULT',
               params='DEFAULT', arguments='DEFAULT', schedule='DEFAULT', no-
               tifications='DEFAULT', repo_http_uri='DEFAULT', repo_ref='DEFAULT',
               remote_host_credential_id='DEFAULT', git_credential_id='DEFAULT',
               docker_image_tag='DEFAULT', cancel_timeout='DEFAULT',
               time_zone='DEFAULT', target_project_id='DEFAULT')
```

Edit a container

Parameters `id` : integer

The ID for the script.

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.
```

docker_command : string

The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]

docker_image_name : string

The name of the docker image to pull from DockerHub.

name : string, optional

The name of the container.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

A definition of the parameters this script accepts **in** the arguments_↵
↵field.

- name : string
The variable's name as used within your code.
- label : string
The label to present to users when asking them **for** the value.
- description : string
A short sentence **or** fragment describing this parameter to the_↵
↵end user.
- type : string
The **type** of parameter. Valid options: string, multi_line_string,
integer, float, bool, file, database, credential_aws,
credential_redshift, **or** credential_custom
- required : boolean
Whether this param **is** required.
- value : string
The value you would like to **set** this param to. Setting this_↵
↵value makes
this parameter a fixed param.
- default : string
If an argument **for** this parameter **is not** defined, it will use_↵
↵this
default value. Use true, **True**, t, y, yes, **or** 1 **for** true bool's_↵
↵**or**
false, **False**, f, n, no, **or** 0 **for** false bool's. Cannot be used_↵
↵**for**
parameters that are required **or** a credential type.

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

- scheduled : boolean
If the **object is** scheduled
- scheduled_days : list
Day based on numeric value starting at 0 **for** Sunday
- scheduled_hours : list
Hours of the day it **is** scheduled on
- scheduled_minutes : list
Minutes of the day it **is** scheduled on
- scheduled_runs_per_hour : integer
Alternative to scheduled minutes, number of times to run per_↵
↵hour

notifications : dict, optional:

- urls : list
URLs to receive a POST request at job completion
- success_email_subject : string
Custom subject line **for** success e-mail.
- success_email_body : string
Custom body text **for** success e-mail, written **in** Markdown.
- success_email_addresses : list
Addresses to notify by e-mail when the job completes_↵
↵successfully.
- success_email_from_name : string

```
    Name from which success emails are sent; defaults to "Civis."  
- success_email_reply_to : string  
    Address for replies to success emails; defaults to the author_  
↳ of the  
    job.  
- failure_email_addresses : list  
    Addresses to notify by e-mail when the job fails.  
- stall_warning_minutes : integer  
    Stall warning emails will be sent after this amount of minutes.  
- success_on : boolean  
    If success email notifications are on.  
- failure_on : boolean  
    If failure email notifications are on.
```

repo_http_uri : string, optional

The location of a github repo to clone into the container, e.g. `github.com/my-user/my-repo.git`.

repo_ref : string, optional

The tag or branch of the github repo to clone into the container.

remote_host_credential_id : integer, optional

The id of the database credentials to pass into the environment of the container.

git_credential_id : integer, optional

The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_image_tag : string, optional

The tag of the docker image to pull from DockerHub (default: latest).

cancel_timeout : integer, optional

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

time_zone : string, optional

The time zone of this script.

target_project_id : integer, optional

Target project to which script outputs will be added.

Returns id : integer

The ID for the script.

name : string

The name of the container.

type : string

The type of the script (e.g Container)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
```

```
The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

template_dependents_count : integer

How many other scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template script.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
```



```

- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
↳ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
↳ of the
  job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

running_as : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

required_resources : dict:

```

- cpu : integer
    The number of CPU shares to allocate for the container. Each,
↳ core has
    1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
↳ This
    space will be used to hold the git repo configured for the,
↳ container
    and anything your container writes to /tmp or /data. Fractional,
↳ values
    (e.g. 0.25) are supported.

```

repo_http_uri : string

The location of a github repo to clone into the container, e.g. `github.com/my-user/my-repo.git`.

repo_ref : string

The tag or branch of the github repo to clone into the container.

remote_host_credential_id : integer

The id of the database credentials to pass into the environment of the container.

git_credential_id : integer

The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_command : string

The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

time_zone : string

The time zone of this script.

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

target_project_id : integer

Target project to which script outputs will be added.

put_containers_archive (*id*, *status*)

Update the archive status of this object

Parameters id : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns id : integer

The ID for the script.

name : string

The name of the container.

type : string

The type of the script (e.g Container)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

template_dependents_count : integer

How many other scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template script.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↳hour

```

notifications : dict:

```

- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↳of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

running_as : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

required_resources : dict:

```

- cpu : integer
    The number of CPU shares to allocate for the container. Each_
↳core has
    1024 shares.
- memory : integer

```

```
The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
  The amount of disk space, in GB, to allocate for the container.
↪This
  space will be used to hold the git repo configured for the
↪container
  and anything your container writes to /tmp or /data. Fractional
↪values
  (e.g. 0.25) are supported.
```

repo_http_uri : string

The location of a github repo to clone into the container, e.g. github.com/my-user/my-repo.git.

repo_ref : string

The tag or branch of the github repo to clone into the container.

remote_host_credential_id : integer

The id of the database credentials to pass into the environment of the container.

git_credential_id : integer

The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_command : string

The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]

docker_image_name : string

The name of the docker image to pull from DockerHub.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

time_zone : string

The time zone of this script.

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

target_project_id : integer

Target project to which script outputs will be added.

put_containers_projects (*id*, *project_id*)

Add a container docker to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_containers_shares_groups (*id*, *group_ids*, *permission_level*,
*, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
```

```
- id : integer
- name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_containers_shares_users (*id*, *user_ids*, *permission_level*,
*, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
```



```
- id : integer
- name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_custom(*id*, *, *name*='DEFAULT', *parent_id*='DEFAULT', *arguments*='DEFAULT', *remote_host_id*='DEFAULT', *credential_id*='DEFAULT', *schedule*='DEFAULT', *notifications*='DEFAULT', *time_zone*='DEFAULT', *target_project_id*='DEFAULT')

Replace all attributes of this Custom Script

Parameters **id** : integer

The ID for the script.

name : string, optional

The name of the script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

credential_id : integer, optional

The credential that this script will use.

schedule : dict, optional:

```
- scheduled : boolean
  If the object is scheduled
- scheduled_days : list
  Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
  Hours of the day it is scheduled on
- scheduled_minutes : list
  Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
  Alternative to scheduled minutes, number of times to run per
  hour
```

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
    successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
    of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

time_zone : string, optional

The time zone of this script.

target_project_id : integer, optional

Target project to which script outputs will be added.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g Custom)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
```

```
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

projects : list:

A `list` of projects containing the script.

```
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

params : list:

A definition of the parameters this script accepts `in` the arguments.

```
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used
↪for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

from_template_id : integer

The ID of the template script.

ui_report_url : integer

The url of the custom HTML.

template_script_name : string

The name of the template script.

template_note : string

The template's note.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

code_preview : string

The code that this script will run with arguments inserted.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
```

```

    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

running_as : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

time_zone : string

The time zone of this script.

last_run : dict:

```

- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

target_project_id : integer

Target project to which script outputs will be added.

put_custom_archive (*id*, *status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **id** : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g Custom)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
```

```

    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.

```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

from_template_id : integer

The ID of the template script.

ui_report_url : integer

The url of the custom HTML.

template_script_name : string

The name of the template script.

template_note : string

The template's note.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

code_preview : string

The code that this script will run with arguments inserted.

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour

```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
↳ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
↳ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

archived : string

The archival status of the requested object(s).

target_project_id : integer

Target project to which script outputs will be added.

put_custom_projects (*id, project_id*)

Add a Job to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_custom_shares_groups (*id, group_ids, permission_level, *, share_email_body='DEFAULT', send_shared_email='DEFAULT'*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_custom_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
 send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns *readers* : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```

- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string

```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_javascript (*id*, *name*, *source*, *remote_host_id*, *credential_id*, *, *parent_id*='DEFAULT', *user_context*='DEFAULT', *params*='DEFAULT', *arguments*='DEFAULT', *schedule*='DEFAULT', *notifications*='DEFAULT', *next_run_at*='DEFAULT', *time_zone*='DEFAULT', *target_project_id*='DEFAULT')

Replace all attributes of this JavaScript Script

Parameters **id** : integer

The ID for the script.

name : string

The name of the script.

source : string

The body/text of the script.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

A definition of the parameters this script accepts **in** the arguments_↵
↵field.

```

- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.
- description : string
  A short sentence or fragment describing this parameter to the_↵
  ↵end user.
- type : string
  The type of parameter. Valid options: string, multi_line_string,

```

```
integer, float, bool, file, database, credential_aws,
credential_redshift, or credential_custom
- required : boolean
  Whether this param is required.
- value : string
  The value you would like to set this param to. Setting this
  ↪value makes
    this parameter a fixed param.
- default : string
  If an argument for this parameter is not defined, it will use
  ↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's
  ↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used
  ↪for
    parameters that are required or a credential type.
```

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

```
- scheduled : boolean
  If the object is scheduled
- scheduled_days : list
  Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
  Hours of the day it is scheduled on
- scheduled_minutes : list
  Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
  Alternative to scheduled minutes, number of times to run per
  ↪hour
```

notifications : dict, optional:

```
- urls : list
  URLs to receive a POST request at job completion
- success_email_subject : string
  Custom subject line for success e-mail.
- success_email_body : string
  Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
  Addresses to notify by e-mail when the job completes
  ↪successfully.
- success_email_from_name : string
  Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
  Address for replies to success emails; defaults to the author
  ↪of the
    job.
- failure_email_addresses : list
  Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
  Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
  If success email notifications are on.
```

```
- failure_on : boolean
    If failure email notifications are on.
```

next_run_at : string/time, optional

The time of the next scheduled run.

time_zone : string, optional

The time zone of this script.

target_project_id : integer, optional

Target project to which script outputs will be added.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
```

```
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

- details : string
The details link to get more information about the script.
- runs : string
The runs link to get the run information **list** **for** this script.

schedule : dict:

- scheduled : boolean
If the **object** **is** scheduled
- scheduled_days : **list**
Day based on numeric value starting at 0 **for** Sunday
- scheduled_hours : **list**
Hours of the day it **is** scheduled on
- scheduled_minutes : **list**
Minutes of the day it **is** scheduled on
- scheduled_runs_per_hour : integer
Alternative to scheduled minutes, number of times to run per_
↳hour

notifications : dict:

- urls : **list**
URLs to receive a POST request at job completion
- success_email_subject : string
Custom subject line **for** success e-mail.
- success_email_body : string
Custom body text **for** success e-mail, written **in** Markdown.
- success_email_addresses : **list**
Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_from_name : string
Name **from** **which** success emails are sent; defaults to "Civis."
- success_email_reply_to : string
Address **for** replies to success emails; defaults to the author_
↳of the
job.
- failure_email_addresses : **list**
Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
If success email notifications are on.
- failure_on : boolean
If failure email notifications are on.

running_as : dict:

- id : integer
The ID of this user.
- name : string
This user's **name**.
- username : string
This user's **username**.
- initials : string
This user's **initials**.

```
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

put_javascript_archive (*id*, *status*)
Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **id** : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
```

```
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
```

```
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

put_javascript_projects (*id*, *project_id*)

Add a scripted sql to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_javascript_shares_groups (*id*, *group_ids*, *permission_level*,
*, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns `readers` : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_javascript_shares_users (*id*, *user_ids*, *permission_level*,
*, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns `readers` : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_python3 (*id*, *name*, *source*, *, *parent_id*='DEFAULT', *user_context*='DEFAULT',
params='DEFAULT', *arguments*='DEFAULT', *schedule*='DEFAULT', *no-*
tifications='DEFAULT', *next_run_at*='DEFAULT', *time_zone*='DEFAULT',
target_project_id='DEFAULT', *required_resources*='DEFAULT', *can-*
cel_timeout='DEFAULT')

Replace all attributes of this Python Script

Parameters **id** : integer

The ID for the script.

name : string

The name of the script.

source : string

The body/text of the script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

A definition of the parameters this script accepts **in** the arguments.
 ↪field.

- name : string
 The variable's name as used within your code.
- label : string
 The label to present to users when asking them **for** the value.
- description : string
 A short sentence **or** fragment describing this parameter to the
 ↪end user.
- type : string
 The **type** of parameter. Valid options: string, multi_line_string,
 integer, float, bool, file, database, credential_aws,
 credential_redshift, **or** credential_custom
- required : boolean
 Whether this param **is** required.
- value : string
 The value you would like to **set** this param to. Setting this
 ↪value makes
 this parameter a fixed param.
- default : string
 If an argument **for** this parameter **is not** defined, it will use
 ↪this
 default value. Use true, **True**, t, y, yes, **or** 1 **for** true bool's.
 ↪or
 false, **False**, f, n, no, **or** 0 **for** false bool's. Cannot be used
 ↪for
 parameters that are required **or** a credential type.

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

- scheduled : boolean
 If the **object is** scheduled
- scheduled_days : list
 Day based on numeric value starting at 0 **for** Sunday
- scheduled_hours : list
 Hours of the day it **is** scheduled on
- scheduled_minutes : list
 Minutes of the day it **is** scheduled on
- scheduled_runs_per_hour : integer
 Alternative to scheduled minutes, number of times to run per
 ↪hour

notifications : dict, optional:

- urls : list
 URLs to receive a POST request at job completion
- success_email_subject : string
 Custom subject line **for** success e-mail.
- success_email_body : string
 Custom body text **for** success e-mail, written **in** Markdown.
- success_email_addresses : list
 Addresses to notify by e-mail when the job completes
 ↪successfully.
- success_email_from_name : string

```
    Name from which success emails are sent; defaults to "Civis."  
- success_email_reply_to : string  
    Address for replies to success emails; defaults to the author_  
↳ of the  
    job.  
- failure_email_addresses : list  
    Addresses to notify by e-mail when the job fails.  
- stall_warning_minutes : integer  
    Stall warning emails will be sent after this amount of minutes.  
- success_on : boolean  
    If success email notifications are on.  
- failure_on : boolean  
    If failure email notifications are on.
```

next_run_at : string/time, optional

The time of the next scheduled run.

time_zone : string, optional

The time zone of this script.

target_project_id : integer, optional

Target project to which script outputs will be added.

required_resources : dict, optional:

```
- cpu : integer  
    The number of CPU shares to allocate for the container. Each_  
↳ core has  
    1024 shares. Must be at least 2 shares.  
- memory : integer  
    The amount of RAM to allocate for the container (in MiB). Must_  
↳ be at  
    least 4 MiB.  
- disk_space : number/float  
    The amount of disk space, in GB, to allocate for the container._  
↳ This  
    space will be used to hold the git repo configured for the_  
↳ container  
    and anything your container writes to /tmp or /data. Fractional_  
↳ values  
    (e.g. 0.25) are supported.
```

cancel_timeout : integer, optional

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
integer, float, bool, file, database, credential_aws,
credential_redshift, or credential_custom
```

```
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```

- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
↳ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
↳ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

running_as : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```

- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must
    ↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.
```

source : string

The body/text of the script.

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

put_python3_archive (*id, status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **id** : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
```

```
The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
```

```

- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
    ↪ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author,
    ↪ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.

```

running_as : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```

- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.

```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must
    ↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.
```

source : string

The body/text of the script.

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

put_python3_projects (*id*, *project_id*)

Add a python docker to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_python3_shares_groups (*id*, *group_ids*, *permission_level*,
*, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_python3_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
 send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_r (*id*, *name*, *source*, *, *parent_id*='DEFAULT', *user_context*='DEFAULT', *params*='DEFAULT', *arguments*='DEFAULT', *schedule*='DEFAULT', *notifications*='DEFAULT', *next_run_at*='DEFAULT', *time_zone*='DEFAULT', *target_project_id*='DEFAULT', *required_resources*='DEFAULT', *cancel_timeout*='DEFAULT')

Replace all attributes of this R Script

Parameters **id** : integer

The ID for the script.

name : string

The name of the script.

source : string

The body/text of the script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

A definition of the parameters this script accepts **in** the arguments_↵
↵field.

- name : string
The variable's name as used within your code.
- label : string
The label to present to users when asking them **for** the value.
- description : string
A short sentence **or** fragment describing this parameter to the_↵
↵end user.
- type : string
The type of parameter. Valid options: string, multi_line_string,
integer, float, bool, file, database, credential_aws,
credential_redshift, **or** credential_custom
- required : boolean
Whether this param **is** required.
- value : string
The value you would like to set this param to. Setting this_↵
↵value makes
this parameter a fixed param.
- default : string
If an argument **for** this parameter **is not** defined, it will use_↵
↵this
default value. Use true, **True**, t, y, yes, **or** 1 **for** true bool's_↵
↵or
false, **False**, f, n, no, **or** 0 **for** false bool's. Cannot be used_↵
↵for
parameters that are required **or** a credential type.

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

- scheduled : boolean
If the object **is** scheduled
- scheduled_days : list
Day based on numeric value starting at 0 **for** Sunday
- scheduled_hours : list
Hours of the day it **is** scheduled on
- scheduled_minutes : list
Minutes of the day it **is** scheduled on
- scheduled_runs_per_hour : integer
Alternative to scheduled minutes, number of times to run per_↵
↵hour

notifications : dict, optional:

- urls : list
URLs to receive a POST request at job completion
- success_email_subject : string
Custom subject line **for** success e-mail.
- success_email_body : string

```
Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
    ↪ successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
    ↪ of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

next_run_at : string/time, optional

The time of the next scheduled run.

time_zone : string, optional

The time zone of this script.

target_project_id : integer, optional

Target project to which script outputs will be added.

required_resources : dict, optional:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each_
    ↪ core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must_
    ↪ be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container._
    ↪ This
    space will be used to hold the git repo configured for the_
    ↪ container
    and anything your container writes to /tmp or /data. Fractional_
    ↪ values
    (e.g. 0.25) are supported.
```

cancel_timeout : integer, optional

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

Returns id : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
```

```
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
    ↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
    ↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
    ↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
```

```
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must
    ↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.
    ↪This
    space will be used to hold the git repo configured for the
    ↪container
    and anything your container writes to /tmp or /data. Fractional
    ↪values
    (e.g. 0.25) are supported.
```

source : string

The body/text of the script.

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

put_r_archive (*id*, *status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **id** : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
```

```
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use_
↪this
    default value. Use true, True, t, y, yes, or 1 for true bool's_
↪or
    false, False, f, n, no, or 0 for false bool's. Cannot be used_
↪for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
```

```
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each_
    ↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must_
    ↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container._
    ↪This
    space will be used to hold the git repo configured for the_
    ↪container
    and anything your container writes to /tmp or /data. Fractional_
    ↪values
    (e.g. 0.25) are supported.
```

source : string

The body/text of the script.

cancel_timeout : integer

The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

put_r_projects (*id*, *project_id*)

Add a r docker to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_r_shares_groups (*id*, *group_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
 send_shared_email='DEFAULT')

Set the permissions groups has on this object

Parameters `id` : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns `readers` : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_r_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters `id` : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns readers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_sql (*id*, *name*, *sql*, *remote_host_id*, *credential_id*, *, *parent_id*=*'DEFAULT'*,
user_context=*'DEFAULT'*, *params*=*'DEFAULT'*, *arguments*=*'DEFAULT'*,
schedule=*'DEFAULT'*, *notifications*=*'DEFAULT'*, *next_run_at*=*'DEFAULT'*,
time_zone=*'DEFAULT'*, *target_project_id*=*'DEFAULT'*, *csv_settings*=*'DEFAULT'*)

Replace all attributes of this SQL script

Parameters id : integer

The ID for the script.

name : string

The name of the script.

sql : string

The raw SQL query for the script.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

parent_id : integer, optional

The ID of the parent job that will trigger this script

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.
- description : string
  A short sentence or fragment describing this parameter to the
↳end user.
- type : string
  The type of parameter. Valid options: string, multi_line_string,
  integer, float, bool, file, database, credential_aws,
  credential_redshift, or credential_custom
- required : boolean
  Whether this param is required.
- value : string
  The value you would like to set this param to. Setting this
↳value makes
  this parameter a fixed param.
- default : string
  If an argument for this parameter is not defined, it will use
↳this
  default value. Use true, True, t, y, yes, or 1 for true bool's
↳or
  false, False, f, n, no, or 0 for false bool's. Cannot be used
↳for
  parameters that are required or a credential type.
```

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

schedule : dict, optional:

```
- scheduled : boolean
  If the object is scheduled
- scheduled_days : list
  Day based on numeric value starting at 0 for Sunday
```

```
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

next_run_at : string/time, optional

The time of the next scheduled run.

time_zone : string, optional

The time zone of this script.

target_project_id : integer, optional

Target project to which script outputs will be added.

csv_settings : dict, optional:

```
- include_header : boolean
    Whether or not to include headers in the output data. Default:_
↪true
- compression : string
    The type of compression to use, if any, one of "none", "zip", or
    "gzip". Default: gzip
- column_delimiter : string
    Which delimiter to use, one of "comma", "tab", or "pipe"._
↪Default:
    comma
- unquoted : boolean
    Whether or not to quote fields. Default: false
```



```

- force_multifile : boolean
    Whether or not the csv should be split into multiple files.
↪Default:
    false
- filename_prefix : string
    A user specified filename prefix for the output file to have.
↪Default:
    null

```

Returns `id` : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```

A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.

```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

A definition of the parameters this script accepts **in** the arguments.

- ↪field.
- name : string
The variable's name as used within your code.
- label : string
The label to present to users when asking them **for** the value.
- description : string
A short sentence **or** fragment describing this parameter to the ↪end user.
- type : string
The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, database, credential_aws, credential_redshift, **or** credential_custom
- required : boolean
Whether this param **is** required.
- value : string
The value you would like to set this param to. Setting this ↪value makes
this parameter a fixed param.
- default : string
If an argument **for** this parameter **is not** defined, it will use ↪this
default value. Use true, **True**, t, y, yes, **or** 1 **for** true bool's ↪or
false, **False**, f, n, no, **or** 0 **for** false bool's. Cannot be used ↪for
parameters that are required **or** a credential type.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

- details : string
The details link to get more information about the script.
- runs : string
The runs link to get the run information **list** **for** this script.

schedule : dict:

- scheduled : boolean
If the **object** **is** scheduled
- scheduled_days : **list**
Day based on numeric value starting at 0 **for** Sunday
- scheduled_hours : **list**
Hours of the day it **is** scheduled on
- scheduled_minutes : **list**
Minutes of the day it **is** scheduled on
- scheduled_runs_per_hour : integer
Alternative to scheduled minutes, number of times to run per_
↪hour

notifications : dict:

- urls : **list**
URLs to receive a POST request at job completion
- success_email_subject : string
Custom subject line **for** success e-mail.
- success_email_body : string
Custom body text **for** success e-mail, written **in** Markdown.
- success_email_addresses : **list**
Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
Name **from** **which** success emails are sent; defaults to "Civis."
- success_email_reply_to : string
Address **for** replies to success emails; defaults to the author_
↪of the
job.
- failure_email_addresses : **list**
Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
If success email notifications are on.
- failure_on : boolean
If failure email notifications are on.

running_as : dict:

- id : integer
The ID of this user.
- name : string
This user's **name**.
- username : string
This user's **username**.
- initials : string
This user's **initials**.
- online : boolean
Whether this user **is** online.

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

sql : string

The raw SQL query for the script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

code_preview : string

The code that this script will run with arguments inserted.

csv_settings : dict:

```
- include_header : boolean
  Whether or not to include headers in the output data. Default: ↵
↵true
- compression : string
  The type of compression to use, if any, one of "none", "zip", or
  "gzip". Default: gzip
- column_delimiter : string
  Which delimiter to use, one of "comma", "tab", or "pipe". ↵
↵Default:
  comma
- unquoted : boolean
```

```

    Whether or not to quote fields. Default: false
- force_multifile : boolean
    Whether or not the csv should be split into multiple files.
↪Default:
    false
- filename_prefix : string
    A user specified filename prefix for the output file to have.
↪Default:
    null

```

put_sql_archive (*id*, *status*)

Update the archive status of this object

Parameters *id* : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns *id* : integer

The ID for the script.

name : string

The name of the script.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

author : dict:

```

- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.

```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

category : string

The category of the script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

parent_id : integer

The ID of the parent job that will trigger this script

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- type : string
    The type of parameter. Valid options: string, multi_line_string,
    integer, float, bool, file, database, credential_aws,
    credential_redshift, or credential_custom
- required : boolean
    Whether this param is required.
- value : string
    The value you would like to set this param to. Setting this
↳value makes
    this parameter a fixed param.
- default : string
    If an argument for this parameter is not defined, it will use
↳this
    default value. Use true, True, t, y, yes, or 1 for true bool's
↳or
    false, False, f, n, no, or 0 for false bool's. Cannot be used
↳for
    parameters that are required or a credential type.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

is_template : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

from_template_id : integer

The ID of the template this script uses, if any.

template_dependents_count : integer

How many other scripts use this one as a template.

template_script_name : string

The name of the template script.

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_from_name : string
    Name from which success emails are sent; defaults to "Civis."
- success_email_reply_to : string
    Address for replies to success emails; defaults to the author_
↪of the
    job.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_on : boolean
    If success email notifications are on.
- failure_on : boolean
    If failure email notifications are on.
```

running_as : dict:

```
- id : integer
    The ID of this user.
- name : string
```

```
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
- finished_at : string/time
    The time that the run completed.
- error : string
    The error message for this run, if present.
```

hidden : boolean

The hidden status of the object.

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

sql : string

The raw SQL query for the script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

remote_host_id : integer

The remote host ID that this script will connect to.

credential_id : integer

The credential that this script will use.

code_preview : string

The code that this script will run with arguments inserted.

csv_settings : dict:


```

- include_header : boolean
    Whether or not to include headers in the output data. Default:↵
↵true
- compression : string
    The type of compression to use, if any, one of "none", "zip", or
    "gzip". Default: gzip
- column_delimiter : string
    Which delimiter to use, one of "comma", "tab", or "pipe".↵
↵Default:
    comma
- unquoted : boolean
    Whether or not to quote fields. Default: false
- force_multifile : boolean
    Whether or not the csv should be split into multiple files.↵
↵Default:
    false
- filename_prefix : string
    A user specified filename prefix for the output file to have.↵
↵Default:
    null

```

put_sql_projects (*id, project_id*)

Add a scripts to a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_sql_shares_groups (*id, group_ids, permission_level, *, share_email_body='DEFAULT', send_shared_email='DEFAULT'*)

Set the permissions groups has on this object

Parameters *id* : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns *readers* : dict:

```

- users : list::
    - id : integer

```

```
- name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_sql_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
 send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
```

```
- id : integer
- name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Search

class Search (*session_kwargs*, *return_type*='civis')

Methods

<code>list(*[, query, type, offset, order, owner, ...])</code>	Perform a search
<code>list_types()</code>	List available search types

```
list (*,      query='DEFAULT',    type='DEFAULT',    offset='DEFAULT',    order='DEFAULT',
      owner='DEFAULT', limit='DEFAULT', archived='DEFAULT')
Perform a search
```

Parameters **query** : string, optional

The search query.

type : string, optional

The type for the search. It accepts a comma-separated list. Valid arguments are listed on the “GET /search/types” endpoint.

offset : integer, optional

The offset for the search results.

order : string, optional

The field on which to order the result set.

owner : string, optional

The owner for the search.

limit : integer, optional

Defaults to 10. Maximum allowed is 50.

archived : string, optional

If specified, return only results with the chosen archived status; either 'true', 'false', or 'all'. Defaults to 'false'.

Returns total_results : integer

The number of items matching the search query.

aggregations : dict

Aggregations by owner and type for the search results.

results : list:

The items returned by the search.

- **score** : number/float
The relevance score **from the** search request.
- **type** : string
The **type** of the item.
- **id** : integer
The ID of the item.
- **name** : string
The name of the item.
- **type_name** : string
The verbose name of the **type**.
- **updated_at** : string/time
The time the item was last updated.
- **owner** : string
The owner of the item.

list_types()

List available search types

Returns type : string

The name of the item type.

Tables

class Tables (*session_kwargs*, *return_type*='civis')

Methods

<code>get(id)</code>	Show basic table info
<code>get_enhancements_cass_ncoa(id, source_table_id)</code>	View the status of a CASS / NCOA table enhancement

Continued on next page

Table 5.30 – continued from previous page

<i>get_enhancements_geocodings</i> (id, source_table_id)	View the status of a geocoding table enhancement
<i>get_enhancements_prepared_matchings</i> (id, ...)	View a prepared matching enhancement
<i>get_enhancements_table_matchings</i> (id, ...)	View a table matching enhancement
<i>list</i> (*[, database_id, schema, name, search, ...])	List tables
<i>list_columns</i> (id, *[, name, limit, page_num, ...])	List columns in the specified table
<i>patch</i> (id, *[, ontology_mapping, description])	Update a table
<i>post</i> (database_id, schema, name, data)	Import a file into a table
<i>post_enhancements_cass_ncoa</i> (source_table_id, *)	Standardize addresses in a table
<i>post_enhancements_geocodings</i> (source_table_id)	Geocode a table
<i>post_enhancements_prepared_matchings</i> (...)	Match person records against a dynamo table prepared by Civis
<i>post_enhancements_table_matchings</i> (...[, ...])	Match person records against an arbitrary Redshift table
<i>post_refresh</i> (id)	Request a refresh for column and table statistics

get (*id*)

Show basic table info

Parameters *id* : integer**Returns** *id* : integer

The ID of the table.

database_id : integer

The ID of the database.

schema : string

The name of the schema containing the table.

name : string

Name of the table.

description : string

The description of the table, as specified by the table owner

is_view : boolean

True if this table represents a view. False if it represents a regular table.

row_count : integer

The number of rows in the table.

column_count : integer

The number of columns in the table.

size_mb : number/float

The size of the table in megabytes.

owner : string

The database username of the table's owner.

distkey : string

The column used as the Amazon Redshift distkey.

sortkeys : string

The column used as the Amazon Redshift sortkey.

refresh_status : string

How up-to-date the table's statistics on row counts, null counts, distinct counts, and values distributions are. One of: refreshing, stale, or current.

last_refresh : string/date-time

The time of the last statistics refresh.

refresh_id : string

The ID of the most recent statistics refresh.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

ontology_mapping : dict

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

columns : list:

```
- name : string
  Name of the column.
- sql_type : string
  SQL type of the column.
- sample_values : list
  A sample of values from the column.
- encoding : string
  The compression encoding for this columnSee: http://docs.aws.amazon.com/redshift/latest/dg/c\_Compression\_encodings.html
- description : string
  The description of the column, as specified by the table owner
- order : integer
  Relative position of the column in the table.
- min_value : string
  Smallest value in the column.
- max_value : string
  Largest value in the column.
- avg_value : number/float
  Average value of the column, where applicable.
- stddev : number/float
  Stddev of the column, where applicable.
```

```

- value_distribution_percent : dict
    A mapping between each value in the column and the percentage
    ↪ of rows
    with that value. Only present for tables with fewer than
    ↪ approximately
    25,000,000 rows and for columns with fewer than twenty distinct
    ↪ values.
- coverage_count : integer
    Number of non-null values in the column.
- null_count : integer
    Number of null values in the column.
- possible_dependent_variable_types : list
    Possible dependent variable types the column may be used to
    ↪ model.
    Null if it may not be used as a dependent variable.
- useable_as_independent_variable : boolean
    Whether the column may be used as an independent variable to
    ↪ train a
    model.
- useable_as_primary_key : boolean
    Whether the column may be used as an primary key to identify
    ↪ table
    rows.
- value_distribution : dict
    An object mapping distinct values in the column to the number
    ↪ of times
    they appear in the column
- distinct_count : integer
    Number of distinct values in the column.

```

joins : list:

```

- id : integer
- left_table_id : integer
- left_identifier : string
- right_table_id : integer
- right_identifier : string
- on : string
- left_join : boolean
- created_at : string/time
- updated_at : string/time

```

multipart_key : list

enhancements : list:

```

- type : string
- created_at : string/time
- updated_at : string/time
- join_id : integer

```

view_def : string

table_def : string

outgoing_table_matches : list:

```

- source_table_id : integer
    Source table

```

```
- target_type : string
    Target type
- target_id : integer
    Target ID
- target : dict::
    - name : string
- job : dict::
    - id : integer
    - name : string
    - type : string
    - state : string
        Whether the job is idle, queued, running, cancelled, or
        failed.
- created_at : string/date-time
- updated_at : string/date-time
- runs : list::
    Information about the most recent runs of the job.
    - id : integer
    - state : string
    - created_at : string/time
        The time that the run was queued.
    - started_at : string/time
        The time that the run started.
    - finished_at : string/time
        The time that the run completed.
    - error : string
        The error message for this run, if present.
- last_run : dict::
    - id : integer
    - state : string
    - created_at : string/time
        The time that the run was queued.
    - started_at : string/time
        The time that the run started.
    - finished_at : string/time
        The time that the run completed.
    - error : string
        The error message for this run, if present.
- hidden : boolean
    The hidden status of the object.
- match_options : dict::
    - max_matches : integer
    - threshold : string
```

get_enhancements_cass_ncoa (*id*, *source_table_id*)

View the status of a CASS / NCOA table enhancement

Parameters *id* : integer

The ID of the enhancement.

source_table_id : integer

The ID of the table that was enhanced.

Returns *id* : integer

The ID of the enhancement.

source_table_id : integer

The ID of the table that was enhanced.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

perform_ncoa : boolean

Whether to update addresses for records matching the National Change of Address (NCOA) database.

ncoa_credential_id : integer

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

output_level : string

The set of fields persisted by a CASS or NCOA enhancement. For CASS enhancements, one of 'cass' or 'all.' For NCOA enhancements, one of 'cass', 'ncoa', 'coalesced' or 'all'. By default, all fields will be returned.

get_enhancements_geocodings (*id*, *source_table_id*)

View the status of a geocoding table enhancement

Parameters **id** : integer

The ID of the enhancement.

source_table_id : integer

The ID of the table that was enhanced.

Returns **id** : integer

The ID of the enhancement.

source_table_id : integer

The ID of the table that was enhanced.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

get_enhancements_prepared_matchings (*id*, *source_table_id*)

View a prepared matching enhancement

Parameters **id** : integer

The ID of the enhancement.

source_table_id : integer

The ID of the table that was enhanced.

Returns id : integer

The ID of the enhancement.

source_table_id : integer

The ID of the table that was enhanced.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match. Must be less than or equal to 1 and greater than or equal to 0.

max_matches : integer

The maximum number of individuals a person may be matched with. A value of 0 indicates that all matches should be returned.

match_table_id : integer

The ID of the Dynamo table to match against.

get_enhancements_table_matchings (*id*, *source_table_id*)

View a table matching enhancement

Parameters id : integer

The ID of the enhancement.

source_table_id : integer

The ID of the table that was enhanced.

Returns id : integer

The ID of the enhancement.

source_table_id : integer

The ID of the table that was enhanced.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match. Must be less than or equal to 1 and greater than or equal to 0.

max_matches : integer

The maximum number of individuals a person may be matched with. A value of 0 indicates that all matches should be returned.

match_table_id : integer

The ID of the Redshift table to match against.

```
list (*, database_id='DEFAULT', schema='DEFAULT', name='DEFAULT', search='DEFAULT',
      limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', iterator='DEFAULT')
List tables
```

Parameters database_id : integer, optional

The ID of the database.

schema : string, optional

If specified, will be used to filter the tables returned. Substring matching is supported with “%” and “*” wildcards (e.g., “schema=%census%” will return both “client_census.table” and “census_2010.table”).

name : string, optional

If specified, will be used to filter the tables returned. Substring matching is supported with “%” and “*” wildcards (e.g., “name=%table%” will return both “table1” and “my table”).

search : string, optional

If specified, will be used to filter the tables returned. Will search across schema and name (in the full form schema.name) and will return any full name containing the search string.

limit : integer, optional

Number of results to return. Defaults to 50. Maximum allowed is 1000.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to schema. Must be one of: schema, name, search.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to asc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns id : integer

The ID of the table.

database_id : integer

The ID of the database.

schema : string

The name of the schema containing the table.

name : string

Name of the table.

description : string

The description of the table, as specified by the table owner

is_view : boolean

True if this table represents a view. False if it represents a regular table.

row_count : integer

The number of rows in the table.

column_count : integer

The number of columns in the table.

size_mb : number/float

The size of the table in megabytes.

owner : string

The database username of the table's owner.

distkey : string

The column used as the Amazon Redshift distkey.

sortkeys : string

The column used as the Amazon Redshift sortkey.

refresh_status : string

How up-to-date the table's statistics on row counts, null counts, distinct counts, and values distributions are. One of: refreshing, stale, or current.

last_refresh : string/date-time

The time of the last statistics refresh.

refresh_id : string

The ID of the most recent statistics refresh.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

```
list_columns (id, *, name='DEFAULT', limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', iterator='DEFAULT')
```

List columns in the specified table

Parameters *id* : integer

name : string, optional

Search for columns with the given name, within the specified table.

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to name. Must be one of: name, order.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to asc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns *name* : string

Name of the column.

sql_type : string

SQL type of the column.

sample_values : list

A sample of values from the column.

encoding : string

The compression encoding for this columnSee: http://docs.aws.amazon.com/redshift/latest/dg/c_Compression_encodings.html

description : string

The description of the column, as specified by the table owner

order : integer

Relative position of the column in the table.

min_value : string

Smallest value in the column.

max_value : string

Largest value in the column.

avg_value : number/float

Average value of the column, where applicable.

stddev : number/float

Stddev of the column, where applicable.

value_distribution_percent : dict

A mapping between each value in the column and the percentage of rows with that value. Only present for tables with fewer than approximately 25,000,000 rows and for columns with fewer than twenty distinct values.

coverage_count : integer

Number of non-null values in the column.

null_count : integer

Number of null values in the column.

possible_dependent_variable_types : list

Possible dependent variable types the column may be used to model. Null if it may not be used as a dependent variable.

useable_as_independent_variable : boolean

Whether the column may be used as an independent variable to train a model.

useable_as_primary_key : boolean

Whether the column may be used as a primary key to identify table rows.

value_distribution : dict

An object mapping distinct values in the column to the number of times they appear in the column

distinct_count : integer

Number of distinct values in the column.

patch (*id*, *, *ontology_mapping*=*'DEFAULT'*, *description*=*'DEFAULT'*)

Update a table

Parameters **id** : integer

The ID of the table.

ontology_mapping : dict, optional

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

description : string, optional

The user-defined description of the table.

Returns **id** : integer

The ID of the table.

database_id : integer

The ID of the database.

schema : string

The name of the schema containing the table.

name : string

Name of the table.

description : string

The description of the table, as specified by the table owner

is_view : boolean

True if this table represents a view. False if it represents a regular table.

row_count : integer

The number of rows in the table.

column_count : integer

The number of columns in the table.

size_mb : number/float

The size of the table in megabytes.

owner : string

The database username of the table's owner.

distkey : string

The column used as the Amazon Redshift distkey.

sortkeys : string

The column used as the Amazon Redshift sortkey.

refresh_status : string

How up-to-date the table's statistics on row counts, null counts, distinct counts, and values distributions are. One of: refreshing, stale, or current.

last_refresh : string/date-time

The time of the last statistics refresh.

refresh_id : string

The ID of the most recent statistics refresh.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

ontology_mapping : dict

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

post (*database_id, schema, name, data*)

Import a file into a table

Parameters **database_id** : integer

The ID of the destination database.

schema : string

The destination schema name.

name : string

The destination table name, without the schema prefix.

data : string

The file to import, uploaded using HTTP multipart.

Returns **database_id** : integer

The ID of the destination database.

schema : string

The destination schema name.

name : string

The destination table name, without the schema prefix.

state : string

The state of the last run.

started_at : string/date-time

The start time of the last run.

finished_at : string/date-time

The end time of the last run.

post_enhancements_cass_ncoa (*source_table_id, *, perform_ncoa='DEFAULT',
ncoa_credential_id='DEFAULT', output_level='DEFAULT'*)

Standardize addresses in a table

Parameters **source_table_id** : integer

The ID of the table to be enhanced.

perform_ncoa : boolean, optional

Whether to update addresses for records matching the National Change of Address (NCOA) database.

ncoa_credential_id : integer, optional

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

output_level : string, optional

The set of fields persisted by a CASS or NCOA enhancement. For CASS enhancements, one of 'cass' or 'all.' For NCOA enhancements, one of 'cass', 'ncoa', 'coalesced' or 'all'. By default, all fields will be returned.

Returns **id** : integer

The ID of the enhancement.

source_table_id : integer

The ID of the table that was enhanced.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

perform_ncoa : boolean

Whether to update addresses for records matching the National Change of Address (NCOA) database.

ncoa_credential_id : integer

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

output_level : string

The set of fields persisted by a CASS or NCOA enhancement. For CASS enhancements, one of 'cass' or 'all.' For NCOA enhancements, one of 'cass', 'ncoa', 'coalesced' or 'all'. By default, all fields will be returned.

post_enhancements_geocodings (*source_table_id*)

Geocode a table

Parameters **source_table_id** : integer

The ID of the table to be enhanced.

Returns **id** : integer

The ID of the enhancement.

source_table_id : integer

The ID of the table that was enhanced.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

post_enhancements_prepared_matchings (*source_table_id*, *threshold*, *match_table_id*, *,
max_matches='DEFAULT')

Match person records against a dynamo table prepared by Civis

Parameters **source_table_id** : integer

The ID of the table to be enhanced.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match. Must be less than or equal to 1 and greater than or equal to 0.

match_table_id : integer

The ID of the Dynamo table to match against.

max_matches : integer, optional

The maximum number of individuals a person may be matched with. A value of 0 indicates that all matches should be returned.

Returns id : integer

The ID of the enhancement.

source_table_id : integer

The ID of the table that was enhanced.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match. Must be less than or equal to 1 and greater than or equal to 0.

max_matches : integer

The maximum number of individuals a person may be matched with. A value of 0 indicates that all matches should be returned.

match_table_id : integer

The ID of the Dynamo table to match against.

post_enhancements_table_matchings (*source_table_id*, *threshold*, *match_table_id*, *,
max_matches='DEFAULT')

Match person records against an arbitrary Redshift table

Parameters source_table_id : integer

The ID of the table to be enhanced.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match. Must be less than or equal to 1 and greater than or equal to 0.

match_table_id : integer

The ID of the Redshift table to match against.

max_matches : integer, optional

The maximum number of individuals a person may be matched with. A value of 0 indicates that all matches should be returned.

Returns id : integer

The ID of the enhancement.

source_table_id : integer

The ID of the table that was enhanced.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match. Must be less than or equal to 1 and greater than or equal to 0.

max_matches : integer

The maximum number of individuals a person may be matched with. A value of 0 indicates that all matches should be returned.

match_table_id : integer

The ID of the Redshift table to match against.

post_refresh (*id*)

Request a refresh for column and table statistics

Parameters id : integer**Returns id** : integer

The ID of the table.

database_id : integer

The ID of the database.

schema : string

The name of the schema containing the table.

name : string

Name of the table.

description : string

The description of the table, as specified by the table owner

is_view : boolean

True if this table represents a view. False if it represents a regular table.

row_count : integer

The number of rows in the table.

column_count : integer

The number of columns in the table.

size_mb : number/float

The size of the table in megabytes.

owner : string

The database username of the table's owner.

distkey : string

The column used as the Amazon Redshift distkey.

sortkeys : string

The column used as the Amazon Redshift sortkey.

refresh_status : string

How up-to-date the table's statistics on row counts, null counts, distinct counts, and values distributions are. One of: refreshing, stale, or current.

last_refresh : string/date-time

The time of the last statistics refresh.

refresh_id : string

The ID of the most recent statistics refresh.

last_run : dict:

```
- id : integer
- state : string
- created_at : string/time
  The time that the run was queued.
- started_at : string/time
  The time that the run started.
- finished_at : string/time
  The time that the run completed.
- error : string
  The error message for this run, if present.
```

ontology_mapping : dict

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

columns : list:

```
- name : string
  Name of the column.
- sql_type : string
  SQL type of the column.
- sample_values : list
  A sample of values from the column.
- encoding : string
  The compression encoding for this columnSee: http://docs.aws.amazon.com/redshift/latest/dg/c\_Compression\_encodings.html
- description : string
  The description of the column, as specified by the table owner
- order : integer
  Relative position of the column in the table.
```

```

- min_value : string
    Smallest value in the column.
- max_value : string
    Largest value in the column.
- avg_value : number/float
    Average value of the column, where applicable.
- stddev : number/float
    Stddev of the column, where applicable.
- value_distribution_percent : dict
    A mapping between each value in the column and the percentage
    ↪ of rows
    with that value. Only present for tables with fewer than
    ↪ approximately
    25,000,000 rows and for columns with fewer than twenty distinct
    ↪ values.
- coverage_count : integer
    Number of non-null values in the column.
- null_count : integer
    Number of null values in the column.
- possible_dependent_variable_types : list
    Possible dependent variable types the column may be used to
    ↪ model.
    Null if it may not be used as a dependent variable.
- useable_as_independent_variable : boolean
    Whether the column may be used as an independent variable to
    ↪ train a
    model.
- useable_as_primary_key : boolean
    Whether the column may be used as an primary key to identify
    ↪ table
    rows.
- value_distribution : dict
    An object mapping distinct values in the column to the number
    ↪ of times
    they appear in the column
- distinct_count : integer
    Number of distinct values in the column.

```

joins : list:

```

- id : integer
- left_table_id : integer
- left_identifier : string
- right_table_id : integer
- right_identifier : string
- on : string
- left_join : boolean
- created_at : string/time
- updated_at : string/time

```

multipart_key : list

enhancements : list:

```

- type : string
- created_at : string/time
- updated_at : string/time
- join_id : integer

```

view_def : string

table_def : string

outgoing_table_matches : list:

```
- source_table_id : integer
    Source table
- target_type : string
    Target type
- target_id : integer
    Target ID
- target : dict::
    - name : string
- job : dict::
    - id : integer
    - name : string
    - type : string
    - state : string
        Whether the job is idle, queued, running, cancelled, or
↳ failed.
- created_at : string/date-time
- updated_at : string/date-time
- runs : list::
    Information about the most recent runs of the job.
    - id : integer
    - state : string
    - created_at : string/time
        The time that the run was queued.
    - started_at : string/time
        The time that the run started.
    - finished_at : string/time
        The time that the run completed.
    - error : string
        The error message for this run, if present.
- last_run : dict::
    - id : integer
    - state : string
    - created_at : string/time
        The time that the run was queued.
    - started_at : string/time
        The time that the run started.
    - finished_at : string/time
        The time that the run completed.
    - error : string
        The error message for this run, if present.
- hidden : boolean
    The hidden status of the object.
- match_options : dict::
    - max_matches : integer
    - threshold : string
```

Templates

class Templates (*session_kwargs*, *return_type*='civis')

Methods

<i>delete_reports_shares_groups</i> (id, group_id)	Revoke the permissions a group has on this object
<i>delete_reports_shares_users</i> (id, user_id)	Revoke the permissions a user has on this object
<i>delete_scripts_projects</i> (id, project_id)	Remove a Template::Script from a project
<i>delete_scripts_shares_groups</i> (id, group_id)	Revoke the permissions a group has on this object
<i>delete_scripts_shares_users</i> (id, user_id)	Revoke the permissions a user has on this object
<i>get_reports</i> (id)	Get a Report Template
<i>get_scripts</i> (id)	Get a Script Template
<i>list_reports</i> (*[, hidden, category, limit, ...])	List Report Templates
<i>list_reports_shares</i> (id)	List users and groups permissioned on this object
<i>list_scripts</i> (*[, hidden, category, limit, ...])	List Script Templates
<i>list_scripts_projects</i> (id, *[, hidden])	List the projects a Template::Script belongs to
<i>list_scripts_shares</i> (id)	List users and groups permissioned on this object
<i>patch_reports</i> (id, *[, name, category, ...])	Update some attributes of this Report Template
<i>patch_scripts</i> (id, *[, name, note, ...])	Update some attributes of this Script Template
<i>post_reports</i> (name, code_body, *[, category, ...])	Create a Report Template
<i>post_scripts</i> (script_id, name, *[, note, ...])	Create a Script Template
<i>put_reports</i> (id, name, code_body, *[, ...])	Replace all attributes of this Report Template
<i>put_reports_shares_groups</i> (id, group_ids, ...)	Set the permissions groups has on this object
<i>put_reports_shares_users</i> (id, user_ids, ...)	Set the permissions users have on this object
<i>put_scripts</i> (id, name, *[, note, ...])	Replace all attributes of this Script Template
<i>put_scripts_projects</i> (id, project_id)	Add a Template::Script to a project
<i>put_scripts_shares_groups</i> (id, group_ids, ...)	Set the permissions groups has on this object
<i>put_scripts_shares_users</i> (id, user_ids, ...)	Set the permissions users have on this object

delete_reports_shares_groups (*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters *id* : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_reports_shares_users (*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters *id* : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

delete_scripts_projects (*id, project_id*)

Remove a Template::Script from a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_scripts_shares_groups (*id, group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_scripts_shares_users (*id, user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get_reports (*id*)

Get a Report Template

Parameters **id** : integer

Returns **id** : integer

name : string

The name of the template.

category : string

The category of this report template. Can be left blank. Acceptable values are: dataset-viz

created_at : string/time

updated_at : string/time

use_count : integer

The number of uses of this template.

archived : boolean

Whether the template has been archived.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

tech_reviewed : boolean

Whether this template has been audited by Civis for security vulnerability and correctness.

auth_code_url : string

A URL to the template's stored code body.

provide_api_key : boolean

Whether reports based on this template request an API Key from the report viewer.

hidden : boolean

The hidden status of the object.

get_scripts (*id*)

Get a Script Template

Parameters *id* : integer

Returns *id* : integer

script_id : integer

The id of the script that this template uses.

user_context : string

The user context of the script that this template uses.

name : string

The name of the template.

category : string

The category of this template.

note : string

A note describing what this template is used for; custom scripts created off this template will display this description.

created_at : string/time

updated_at : string/time

use_count : integer

The number of uses of this template.

ui_report_id : integer

The id of the report that this template uses.

tech_reviewed : boolean

Whether this template has been audited by Civis for security vulnerability and correctness.

archived : boolean

Whether the template has been archived.

hidden : boolean

The hidden status of the object.

```
list_reports (*,             hidden='DEFAULT',      category='DEFAULT',      limit='DEFAULT',  
                page_num='DEFAULT',    order='DEFAULT',    order_dir='DEFAULT',    itera-  
                tor='DEFAULT')
```

List Report Templates

Parameters **hidden** : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

category : string, optional

A category to filter results by, one of: dataset-viz

limit : integer, optional

Number of results to return. Defaults to 50. Maximum allowed is 1000.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to name. Must be one of: name, updated_at, created_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to asc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

name : string

The name of the template.

category : string

The category of this report template. Can be left blank. Acceptable values are: dataset-viz

created_at : string/time

updated_at : string/time

use_count : integer

The number of uses of this template.

archived : boolean

Whether the template has been archived.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

tech_reviewed : boolean

Whether this template has been audited by Civis for security vulnerability and correctness.

list_reports_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **readers** : dict:

```
- users : list::
    - id : integer
    - name : string
- groups : list::
    - id : integer
    - name : string
```

writers : dict:

```
- users : list::
    - id : integer
    - name : string
- groups : list::
    - id : integer
    - name : string
```

owners : dict:

```
- users : list::
    - id : integer
    - name : string
- groups : list::
```

```
- id : integer
- name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_scripts (*, *hidden='DEFAULT', category='DEFAULT', limit='DEFAULT',
page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', itera-
tor='DEFAULT'*)

List Script Templates

Parameters **hidden** : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

category : string, optional

A category to filter results by, one of: import, export, enhancement, model, and script

limit : integer, optional

Number of results to return. Defaults to 50. Maximum allowed is 1000.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to name. Must be one of: name, updated_at, created_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to asc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

script_id : integer

The id of the script that this template uses.

user_context : string

The user context of the script that this template uses.

name : string

The name of the template.

category : string

The category of this template.

created_at : string/time

updated_at : string/time

use_count : integer

The number of uses of this template.

ui_report_id : integer

The id of the report that this template uses.

tech_reviewed : boolean

Whether this template has been audited by Civis for security vulnerability and correctness.

archived : boolean

Whether the template has been archived.

list_scripts_projects (*id*, *, *hidden*=*'DEFAULT'*)

List the projects a Template::Script belongs to

Parameters **id** : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns **id** : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
```

```
This user's username.  
- initials : string  
  This user's initials.  
- online : boolean  
  Whether this user is online.
```

auto_share : boolean

created_at : string/time

updated_at : string/time

archived : string

The archival status of the requested object(s).

list_scripts_shares (*id*)

List users and groups permissioned on this object

Parameters *id* : integer

The ID of the object.

Returns `readers` : dict:

```
- users : list::  
  - id : integer  
  - name : string  
- groups : list::  
  - id : integer  
  - name : string
```

writers : dict:

```
- users : list::  
  - id : integer  
  - name : string  
- groups : list::  
  - id : integer  
  - name : string
```

owners : dict:

```
- users : list::  
  - id : integer  
  - name : string  
- groups : list::  
  - id : integer  
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch_reports (*id*, *, *name*='DEFAULT', *category*='DEFAULT', *archived*='DEFAULT',
code_body='DEFAULT', *provide_api_key*='DEFAULT')

Update some attributes of this Report Template

Parameters *id* : integer

name : string, optional

The name of the template.

category : string, optional

The category of this report template. Can be left blank. Acceptable values are: dataset-viz

archived : boolean, optional

Whether the template has been archived.

code_body : string, optional

The code for the Template body.

provide_api_key : boolean, optional

Whether reports based on this template request an API Key from the report viewer.

Returns *id* : integer

name : string

The name of the template.

category : string

The category of this report template. Can be left blank. Acceptable values are: dataset-viz

created_at : string/time

updated_at : string/time

use_count : integer

The number of uses of this template.

archived : boolean

Whether the template has been archived.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

tech_reviewed : boolean

Whether this template has been audited by Civis for security vulnerability and correctness.

auth_code_url : string

A URL to the template's stored code body.

provide_api_key : boolean

Whether reports based on this template request an API Key from the report viewer.

hidden : boolean

The hidden status of the object.

patch_scripts (*id*, *, *name*='DEFAULT', *note*='DEFAULT', *ui_report_id*='DEFAULT',
archived='DEFAULT')

Update some attributes of this Script Template

Parameters **id** : integer

name : string, optional

The name of the template.

note : string, optional

A note describing what this template is used for; custom scripts created off this template will display this description.

ui_report_id : integer, optional

The id of the report that this template uses.

archived : boolean, optional

Whether the template has been archived.

Returns **id** : integer

script_id : integer

The id of the script that this template uses.

user_context : string

The user context of the script that this template uses.

name : string

The name of the template.

category : string

The category of this template.

note : string

A note describing what this template is used for; custom scripts created off this template will display this description.

created_at : string/time

updated_at : string/time

use_count : integer

The number of uses of this template.

ui_report_id : integer

The id of the report that this template uses.

tech_reviewed : boolean

Whether this template has been audited by Civis for security vulnerability and correctness.

archived : boolean

Whether the template has been archived.

hidden : boolean

The hidden status of the object.

post_reports (*name*, *code_body*, *, *category*='DEFAULT', *archived*='DEFAULT', *provide_api_key*='DEFAULT', *hidden*='DEFAULT')

Create a Report Template

Parameters **name** : string

The name of the template.

code_body : string

The code for the Template body.

category : string, optional

The category of this report template. Can be left blank. Acceptable values are: dataset-viz

archived : boolean, optional

Whether the template has been archived.

provide_api_key : boolean, optional

Whether reports based on this template request an API Key from the report viewer.

hidden : boolean, optional

The hidden status of the object.

Returns **id** : integer

name : string

The name of the template.

category : string

The category of this report template. Can be left blank. Acceptable values are: dataset-viz

created_at : string/time

updated_at : string/time

use_count : integer

The number of uses of this template.

archived : boolean

Whether the template has been archived.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
```

```
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

tech_reviewed : boolean

Whether this template has been audited by Civis for security vulnerability and correctness.

auth_code_url : string

A URL to the template's stored code body.

provide_api_key : boolean

Whether reports based on this template request an API Key from the report viewer.

hidden : boolean

The hidden status of the object.

post_scripts (*script_id*, *name*, *, *note*='DEFAULT', *ui_report_id*='DEFAULT',
archived='DEFAULT', *hidden*='DEFAULT')

Create a Script Template

Parameters **script_id** : integer

The id of the script that this template uses.

name : string

The name of the template.

note : string, optional

A note describing what this template is used for; custom scripts created off this template will display this description.

ui_report_id : integer, optional

The id of the report that this template uses.

archived : boolean, optional

Whether the template has been archived.

hidden : boolean, optional

The hidden status of the object.

Returns **id** : integer

script_id : integer

The id of the script that this template uses.

user_context : string

The user context of the script that this template uses.

name : string

The name of the template.

category : string

The category of this template.

note : string

A note describing what this template is used for; custom scripts created off this template will display this description.

created_at : string/time

updated_at : string/time

use_count : integer

The number of uses of this template.

ui_report_id : integer

The id of the report that this template uses.

tech_reviewed : boolean

Whether this template has been audited by Civis for security vulnerability and correctness.

archived : boolean

Whether the template has been archived.

hidden : boolean

The hidden status of the object.

put_reports (*id*, *name*, *code_body*, *, *category*=*'DEFAULT'*, *archived*=*'DEFAULT'*, *provide_api_key*=*'DEFAULT'*)

Replace all attributes of this Report Template

Parameters id : integer

name : string

The name of the template.

code_body : string

The code for the Template body.

category : string, optional

The category of this report template. Can be left blank. Acceptable values are: dataset-viz

archived : boolean, optional

Whether the template has been archived.

provide_api_key : boolean, optional

Whether reports based on this template request an API Key from the report viewer.

Returns id : integer

name : string

The name of the template.

category : string

The category of this report template. Can be left blank. Acceptable values are: dataset-viz

created_at : string/time

updated_at : string/time

use_count : integer

The number of uses of this template.

archived : boolean

Whether the template has been archived.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

tech_reviewed : boolean

Whether this template has been audited by Civis for security vulnerability and correctness.

auth_code_url : string

A URL to the template's stored code body.

provide_api_key : boolean

Whether reports based on this template request an API Key from the report viewer.

hidden : boolean

The hidden status of the object.

put_reports_shares_groups (*id*, *group_ids*, *permission_level*,
*, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns `readers` : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_reports_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT',
send_shared_email='DEFAULT')

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns `readers` : dict:

```
- users : list::  
  - id : integer  
  - name : string  
- groups : list::  
  - id : integer  
  - name : string
```

writers : dict:

```
- users : list::  
  - id : integer  
  - name : string  
- groups : list::  
  - id : integer  
  - name : string
```

owners : dict:

```
- users : list::  
  - id : integer  
  - name : string  
- groups : list::  
  - id : integer  
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_scripts (*id*, *name*, *, *note*='DEFAULT', *ui_report_id*='DEFAULT', *archived*='DEFAULT')

Replace all attributes of this Script Template

Parameters **id** : integer

name : string

The name of the template.

note : string, optional

A note describing what this template is used for; custom scripts created off this template will display this description.

ui_report_id : integer, optional

The id of the report that this template uses.

archived : boolean, optional

Whether the template has been archived.

Returns **id** : integer

script_id : integer

The id of the script that this template uses.

user_context : string

The user context of the script that this template uses.

name : string

The name of the template.

category : string

The category of this template.

note : string

A note describing what this template is used for; custom scripts created off this template will display this description.

created_at : string/time

updated_at : string/time

use_count : integer

The number of uses of this template.

ui_report_id : integer

The id of the report that this template uses.

tech_reviewed : boolean

Whether this template has been audited by Civis for security vulnerability and correctness.

archived : boolean

Whether the template has been archived.

hidden : boolean

The hidden status of the object.

put_scripts_projects (*id, project_id*)

Add a Template::Script to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_scripts_shares_groups (*id, group_ids, permission_level, *, share_email_body='DEFAULT', send_shared_email='DEFAULT'*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns readers : dict:

```
- users : list::
    - id : integer
    - name : string
- groups : list::
    - id : integer
    - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

```
put_scripts_shares_users(id, user_ids, permission_level, *, share_email_body='DEFAULT',
                          send_shared_email='DEFAULT')
```

Set the permissions users have on this object

Parameters `id` : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Users

class **Users** (*session_kwargs*, *return_type*='civis')

Methods

<code>delete_api_keys(id, key_id)</code>	Revoke the specified API key
<code>get(id)</code>	Show info about a user
<code>get_api_keys(id, key_id)</code>	Show the specified API key
<code>list(*[, feature_flag, account_status, ...])</code>	List users
<code>list_api_keys(id, *[, limit, page_num, ...])</code>	Show API keys belonging to the specified user
<code>list_me()</code>	Show info about the logged-in user

Continued on next page

Table 5.32 – continued from previous page

<code>patch_me(*[, preferences, ...])</code>	Update info about the logged-in user
<code>post_api_keys(id, name, expires_in, *[, ...])</code>	Create a new API key belonging to the logged-in user

delete_api_keys (*id*, *key_id*)

Revoke the specified API key

Parameters *id* : string

The ID of the user or ‘me’.

key_id : integer

The ID of the API key.

Returns *id* : integer

The ID of the API key.

name : string

The name of the API key.

expires_at : string/date-time

The date and time when the key expired.

created_at : string/date-time

The date and time when the key was created.

revoked_at : string/date-time

The date and time when the key was revoked.

last_used_at : string/date-time

The date and time when the key was last used.

scopes : list

The scopes which the key is permissioned on.

use_count : integer

The number of times the key has been used.

expired : boolean

True if the key has expired.

active : boolean

True if the key has neither expired nor been revoked.

constraints : list:

Constraints on the abilities of the created key

- **constraint** : string
The path matcher of the constraint.
- **constraint_type** : string
The **type** of constraint (exact/prefix/regex/verb).
- **get_allowed** : boolean
Whether the constraint allows GET requests.
- **head_allowed** : boolean
Whether the constraint allows HEAD requests.
- **post_allowed** : boolean

```

    Whether the constraint allows POST requests.
- put_allowed : boolean
    Whether the constraint allows PUT requests.
- patch_allowed : boolean
    Whether the constraint allows PATCH requests.
- delete_allowed : boolean
    Whether the constraint allows DELETE requests.

```

get (*id*)

Show info about a user

Parameters *id* : integer

The ID of this user.

Returns *id* : integer

The ID of this user.

user : string

The username of this user.

name : string

The name of this user.

email : string

The email of this user.

active : string

The account status of this user.

primary_group_id : integer

The ID of the primary group of this user.

groups : list:

```

An array of all the groups this user is in.
- id : integer
    The ID of this group.
- name : string
    The name of this group.
- organization_id : integer
    The organization associated with this group.

```

city : string

The city of this user.

state : string

The state of this user.

time_zone : string

The time zone of this user.

initials : string

The initials of this user.

department : string

The department of this user.

title : string

The title of this user.

github_username : string

The GitHub username of this user.

prefers_sms_otp : string

The preference for phone authorization of this user

vpn_enabled : string

The availability of vpn for this user.

otp_required_for_login : string

The two factor authorization requirement for this user.

phone : string

The phone number of this user.

get_api_keys (*id*, *key_id*)

Show the specified API key

Parameters **id** : string

The ID of the user or 'me'.

key_id : integer

The ID of the API key.

Returns **id** : integer

The ID of the API key.

name : string

The name of the API key.

expires_at : string/date-time

The date and time when the key expired.

created_at : string/date-time

The date and time when the key was created.

revoked_at : string/date-time

The date and time when the key was revoked.

last_used_at : string/date-time

The date and time when the key was last used.

scopes : list

The scopes which the key is permissioned on.

use_count : integer

The number of times the key has been used.

expired : boolean

True if the key has expired.

active : boolean

True if the key has neither expired nor been revoked.

constraints : list:

```
Constraints on the abilities of the created key
- constraint : string
    The path matcher of the constraint.
- constraint_type : string
    The type of constraint (exact/prefix/regex/verb).
- get_allowed : boolean
    Whether the constraint allows GET requests.
- head_allowed : boolean
    Whether the constraint allows HEAD requests.
- post_allowed : boolean
    Whether the constraint allows POST requests.
- put_allowed : boolean
    Whether the constraint allows PUT requests.
- patch_allowed : boolean
    Whether the constraint allows PATCH requests.
- delete_allowed : boolean
    Whether the constraint allows DELETE requests.
```

```
list (*,      feature_flag='DEFAULT',      account_status='DEFAULT',      query='DEFAULT',
      group_id='DEFAULT',      organization_id='DEFAULT',      exclude_groups='DEFAULT',
      limit='DEFAULT',      page_num='DEFAULT',      order='DEFAULT',      order_dir='DEFAULT',
      iterator='DEFAULT')
List users
```

Parameters **feature_flag** : string, optional

Return users that have a feature flag enabled.

account_status : string, optional

The account status by which to filter users. May be one of “active”, “inactive”, or “all”.

query : string, optional

Return users who match the given query, based on name, user, and email.

group_id : integer, optional

The ID of the group by which to filter users. Cannot be present if organization_id is.

organization_id : integer, optional

The ID of the organization by which to filter users. Cannot be present if group_id is.

exclude_groups : boolean, optional

Whether or to exclude users’ groups. Default: false.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 10000.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to name. Must be one of: name, user.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to asc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns id : integer

The ID of this user.

user : string

The username of this user.

name : string

The name of this user.

email : string

The email of this user.

active : string

The account status of this user.

primary_group_id : integer

The ID of the primary group of this user.

groups : list:

An array of **all** the groups this user **is in**.

- **id** : integer
The ID of this group.
- **name** : string
The name of this group.
- **organization_id** : integer
The organization associated **with** this group.

created_at : string/date-time

The date and time when the user was created.

current_sign_in_at : string/date-time

The date and time when the user's current session began.

list_api_keys (*id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

Show API keys belonging to the specified user

Parameters id : string

The ID of the user or 'me'.

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns id : integer

The ID of the API key.

name : string

The name of the API key.

expires_at : string/date-time

The date and time when the key expired.

created_at : string/date-time

The date and time when the key was created.

revoked_at : string/date-time

The date and time when the key was revoked.

last_used_at : string/date-time

The date and time when the key was last used.

scopes : list

The scopes which the key is permissioned on.

use_count : integer

The number of times the key has been used.

expired : boolean

True if the key has expired.

active : boolean

True if the key has neither expired nor been revoked.

constraint_count : integer

The number of constraints on the created key

list_me()

Show info about the logged-in user

Returns id : integer

The ID of this user.

name : string

This user's name.

email : string

This user's email address.

username : string

This user's username.

initials : string

This user's initials.

last_checked_announcements : string/date-time

The date and time at which the user last checked their announcements.

feature_flags : dict

The feature flag settings for this user.

roles : list

The roles this user has, listed by slug.

preferences : dict

This user's preferences.

custom_branding : string

The branding of Platform for this user.

groups : list:

An array of **all** the groups this user **is in**.

- **id** : integer
The ID of this group.
- **name** : string
The name of this group.
- **organization_id** : integer
The organization associated **with** this group.

organization_name : string

The name of the organization the user belongs to.

created_at : string/date-time

The date and time when the user was created.

sign_in_count : integer

The number of times the user has signed in.

patch_me (*, preferences='DEFAULT', last_checked_announcements='DEFAULT')

Update info about the logged-in user

Parameters preferences : dict, optional:

- **app_index_order_field** : string
Order field **for** the apps index pages.
- **app_index_order_dir** : string
Order direction **for** the apps index pages.
- **result_index_order_field** : string
Order field **for** the results index page.


```

- result_index_order_dir : string
    Order direction for the results index page.
- result_index_type_filter : string
    Type filter for the results index page.
- result_index_author_filter : string
    Author filter for the results index page.
- result_index_archived_filter : string
    Archived filter for the results index page.
- import_index_order_field : string
    Order field for the imports index page.
- import_index_order_dir : string
    Order direction for the imports index page.
- import_index_type_filter : string
    Type filter for the imports index page.
- import_index_author_filter : string
    Author filter for the imports index page.
- import_index_dest_filter : string
    Destination filter for the imports index page.
- import_index_status_filter : string
    Status filter for the imports index page.
- import_index_archived_filter : string
    Archived filter for the imports index page.
- export_index_order_field : string
    Order field for the exports index page.
- export_index_order_dir : string
    Order direction for the exports index page.
- export_index_type_filter : string
    Type filter for the exports index page.
- export_index_author_filter : string
    Author filter for the exports index page.
- export_index_status_filter : string
    Status filter for the exports index page.
- model_index_order_field : string
    Order field for the models index page.
- model_index_order_dir : string
    Order direction for the models index page.
- model_index_author_filter : string
    Author filter for the models index page.
- model_index_status_filter : string
    Status filter for the models index page.
- model_index_archived_filter : string
    Archived filter for the models index page.
- model_index_thumbnail_view : string
    Thumbnail view for the models index page.
- script_index_order_field : string
    Order field for the scripts index page.
- script_index_order_dir : string
    Order direction for the scripts index page.
- script_index_type_filter : string
    Type filter for the scripts index page.
- script_index_author_filter : string
    Author filter for the scripts index page.
- script_index_status_filter : string
    Status filter for the scripts index page.
- script_index_archived_filter : string
    Archived filter for the scripts index page.
- project_index_order_field : string
    Order field for the projects index page.

```

```
- project_index_order_dir : string
    Order direction for the projects index page.
- project_index_author_filter : string
    Author filter for the projects index page.
- project_index_archived_filter : string
    Archived filter for the projects index page.
- report_index_thumbnail_view : string
    Thumbnail view for the reports index page.
- project_detail_order_field : string
    Order field for projects detail pages.
- project_detail_order_dir : string
    Order direction for projects detail pages.
- project_detail_author_filter : string
    Author filter for projects detail pages.
- project_detail_type_filter : string
    Type filter for projects detail pages.
- project_detail_archived_filter : string
    Archived filter for the projects detail pages.
- enhancement_index_order_field : string
    Order field for the enhancements index page.
- enhancement_index_order_dir : string
    Order direction for the enhancements index page.
- enhancement_index_author_filter : string
    Author filter for the enhancements index page.
- enhancement_index_archived_filter : string
    Archived filter for the enhancements index page.
- preferred_server_id : integer
    ID of preferred server.
- civis_explore_skip_intro : boolean
    Whether the user is shown steps for each exploration.
- registration_index_order_field : string
    Order field for the registrations index page.
- registration_index_order_dir : string
    Order direction for the registrations index page.
- registration_index_status_filter : string
    Status filter for the registrations index page.
- upgrade_requested : string
    Whether a free trial upgrade has been requested.
- welcome_order_field : string
    Order direction for the welcome page.
- welcome_order_dir : string
    Order direction for the welcome page.
- welcome_author_filter : string
    Status filter for the welcome page.
- welcome_status_filter : string
    Status filter for the welcome page.
- welcome_archived_filter : string
    Status filter for the welcome page.
- data_pane_width : string
    Width of the data pane when expanded.
- data_pane_collapsed : string
    Whether the data pane is collapsed.
- notebook_order_field : string
    Order field for the notebooks page.
- notebook_order_dir : string
    Order direction for the notebooks page.
- notebook_author_filter : string
    Author filter for the notebooks page.
```

```

- notebook_archived_filter : string
  Archived filter for the notebooks page.
- notebook_status_filter : string
  Status filter for the notebooks page.
- workflow_index_order_field : string
  Order field for the workflows page.
- workflow_index_order_dir : string
  Order direction for the workflows page.
- workflow_index_author_filter : string
  Author filter for the workflows page.

```

last_checked_announcements : string/date-time, optional

The date and time at which the user last checked their announcements.

Returns id : integer

The ID of this user.

name : string

This user's name.

email : string

This user's email address.

username : string

This user's username.

initials : string

This user's initials.

last_checked_announcements : string/date-time

The date and time at which the user last checked their announcements.

feature_flags : dict

The feature flag settings for this user.

roles : list

The roles this user has, listed by slug.

preferences : dict

This user's preferences.

custom_branding : string

The branding of Platform for this user.

groups : list:

```

An array of all the groups this user is in.
- id : integer
  The ID of this group.
- name : string
  The name of this group.
- organization_id : integer
  The organization associated with this group.

```

organization_name : string

The name of the organization the user belongs to.

created_at : string/date-time

The date and time when the user was created.

sign_in_count : integer

The number of times the user has signed in.

post_api_keys (*id, name, expires_in, *, constraints='DEFAULT'*)

Create a new API key belonging to the logged-in user

Parameters **id** : string

The ID of the user or 'me'.

name : string

The name of the API key.

expires_in : integer

The number of seconds the key should last for.

constraints : list, optional:

Constraints on the abilities of the created key.

- **constraint** : string
The path matcher of the constraint.
- **constraint_type** : string
The **type** of constraint (exact/prefix/regex/verb).
- **get_allowed** : boolean
Whether the constraint allows GET requests.
- **head_allowed** : boolean
Whether the constraint allows HEAD requests.
- **post_allowed** : boolean
Whether the constraint allows POST requests.
- **put_allowed** : boolean
Whether the constraint allows PUT requests.
- **patch_allowed** : boolean
Whether the constraint allows PATCH requests.
- **delete_allowed** : boolean
Whether the constraint allows DELETE requests.

Returns **id** : integer

The ID of the API key.

name : string

The name of the API key.

expires_at : string/date-time

The date and time when the key expired.

created_at : string/date-time

The date and time when the key was created.

revoked_at : string/date-time

The date and time when the key was revoked.

last_used_at : string/date-time

The date and time when the key was last used.

scopes : list

The scopes which the key is permissioned on.

use_count : integer

The number of times the key has been used.

expired : boolean

True if the key has expired.

active : boolean

True if the key has neither expired nor been revoked.

constraints : list:

```
Constraints on the abilities of the created key
- constraint : string
    The path matcher of the constraint.
- constraint_type : string
    The type of constraint (exact/prefix/regex/verb).
- get_allowed : boolean
    Whether the constraint allows GET requests.
- head_allowed : boolean
    Whether the constraint allows HEAD requests.
- post_allowed : boolean
    Whether the constraint allows POST requests.
- put_allowed : boolean
    Whether the constraint allows PUT requests.
- patch_allowed : boolean
    Whether the constraint allows PATCH requests.
- delete_allowed : boolean
    Whether the constraint allows DELETE requests.
```

token : string

The API key.

Workflows

class Workflows (*session_kwargs*, *return_type*='civis')

Methods

<code>delete_projects(id, project_id)</code>	Remove a Workflow::Workflow from a project
<code>delete_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>get(id)</code>	Get a Workflow
<code>get_executions(id, execution_id)</code>	Get a workflow execution
<code>list(*[, hidden, archived, author, limit, ...])</code>	List Workflows
<code>list_executions(id, *[, limit, page_num, ...])</code>	List workflow executions
<code>list_projects(id, *[, hidden])</code>	List the projects a Workflow::Workflow belongs to
<code>list_shares(id)</code>	List users and groups permissioned on this object

Continued on next page

Table 5.33 – continued from previous page

<i>patch</i> (id, *[, name, definition, schedule, ...])	Update some attributes of this Workflow
<i>post</i> (name, *[, from_job_chain, definition, ...])	Create a Workflow
<i>post_clone</i> (id, *[, clone_schedule, ...])	Clone this workflow
<i>post_executions</i> (id, *[, target_task])	Execute a workflow
<i>post_executions_cancel</i> (id, execution_id)	Cancel a workflow execution
<i>post_executions_resume</i> (id, execution_id)	Resume a paused workflow execution
<i>post_executions_retry</i> (id, execution_id, *[, ...])	Retry a failed task, or all failed tasks in an execution
<i>put</i> (id, name, *[, definition, schedule, ...])	Replace all attributes of this Workflow
<i>put_archive</i> (id, status)	Update the archive status of this object
<i>put_projects</i> (id, project_id)	Add a Workflow::Workflow to a project
<i>put_shares_groups</i> (id, group_ids, ...[, ...])	Set the permissions groups has on this object
<i>put_shares_users</i> (id, user_ids, ...[, ...])	Set the permissions users have on this object

delete_projects (*id*, *project_id*)

Remove a Workflow::Workflow from a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_shares_groups (*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters *id* : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users (*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters *id* : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get (*id*)

Get a Workflow

Parameters *id* : integer

Returns id : integer

The ID for this workflow.

name : string

The name of this workflow.

definition : string

The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.

valid : boolean

The validity of the workflow definition.

validation_errors : string

The errors encountered when validating the workflow definition.

file_id : string

The file id for the s3 file containing the workflow configuration.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The state of the workflow. State is “running” if any execution is running, otherwise reflects most recent execution state.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
    hour
```

time_zone : string

The time zone of this workflow.

next_execution_at : string/time

The time of the next scheduled execution.

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- failure_on : boolean
    If failure email notifications are on
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

created_at : string/time

updated_at : string/time

get_executions (*id*, *execution_id*)

Get a workflow execution

Parameters **id** : integer

The ID for the workflow.

execution_id : integer

The ID for the workflow execution.

Returns **id** : integer

The ID for this workflow execution.

state : string

The state of this workflow execution.

mistral_state : string

The state of this workflow as reported by mistral. One of running, paused, success, error, or cancelled

mistral_state_info : string

The state info of this workflow as reported by mistral.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
```



```

    This user's username.
-   initials : string
    This user's initials.
-   online : boolean
    Whether this user is online.

```

definition : string

The definition of the workflow for this execution.

tasks : list:

```

The tasks associated with this execution.
-   name : string
    The name of the task.
-   mistral_state : string
    The state of this task. One of idle, waiting, running, running_
    ↪delayed,
    success, or error
-   mistral_state_info : string
    Extra info associated with the state of the task.
-   runs : list::
    The runs associated with this task, in descending order by id.
    -   id : integer
        The ID of the run.
    -   job_id : integer
        The ID of the job associated with the run.

```

started_at : string/time

The time this execution started.

finished_at : string/time

The time this execution finished.

created_at : string/time

The time this execution was created.

updated_at : string/time

The time this execution was last updated.

list (*, *hidden*='DEFAULT', *archived*='DEFAULT', *author*='DEFAULT', *limit*='DEFAULT',
page_num='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')
 List Workflows

Parameters **hidden** : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

archived : string, optional

The archival status of the requested object(s).

author : string, optional

If specified, return workflows from this author. It accepts a comma- separated list of author ids.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to `updated_at`. Must be one of: `updated_at`, `name`, `created_at`.

order_dir : string, optional

Direction in which to sort, either `asc` (ascending) or `desc` (descending) defaulting to `desc`.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by `limit` are needed. When True, `limit` and `page_num` are ignored. Defaults to False.

Returns **id** : integer

The ID for this workflow.

name : string

The name of this workflow.

valid : boolean

The validity of the workflow definition.

file_id : string

The file id for the s3 file containing the workflow configuration.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The state of the workflow. State is “running” if any execution is running, otherwise reflects most recent execution state.

time_zone : string

The time zone of this workflow.

next_execution_at : string/time

The time of the next scheduled execution.

archived : string

The archival status of the requested object(s).

created_at : string/time

updated_at : string/time

list_executions (*id*, *, *limit*='DEFAULT', *page_num*='DEFAULT', *order*='DEFAULT', *order_dir*='DEFAULT', *iterator*='DEFAULT')

List workflow executions

Parameters **id** : integer

The ID for this workflow.

limit : integer, optional

Number of results to return. Defaults to 20. Maximum allowed is 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id, updated_at, created_at.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **id** : integer

The ID for this workflow execution.

state : string

The state of this workflow execution.

mistral_state : string

The state of this workflow as reported by mistral. One of running, paused, success, error, or cancelled

mistral_state_info : string

The state info of this workflow as reported by mistral.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

started_at : string/time

The time this execution started.

finished_at : string/time

The time this execution finished.

created_at : string/time

The time this execution was created.

updated_at : string/time

The time this execution was last updated.

list_projects (*id*, *, *hidden*='DEFAULT')

List the projects a Workflow::Workflow belongs to

Parameters **id** : integer

The ID of the resource.

hidden : boolean, optional

If specified to be true, returns hidden objects. Defaults to false, returning non-hidden objects.

Returns **id** : integer

The ID for this project.

author : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

name : string

The name of this project.

description : string

A description of the project

users : list:

```
Users who can see the project
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

auto_share : boolean
created_at : string/time
updated_at : string/time
archived : string

The archival status of the requested object(s).

list_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch (*id*, *, *name*='DEFAULT', *definition*='DEFAULT', *schedule*='DEFAULT',
time_zone='DEFAULT', *notifications*='DEFAULT')

Update some attributes of this Workflow

Parameters **id** : integer

The ID for this workflow.

name : string, optional

The name of this workflow.

definition : string, optional

The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

time_zone : string, optional

The time zone of this workflow.

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- failure_on : boolean
    If failure email notifications are on
```

Returns id : integer

The ID for this workflow.

name : string

The name of this workflow.

definition : string

The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.

valid : boolean

The validity of the workflow definition.

validation_errors : string

The errors encountered when validating the workflow definition.

file_id : string

The file id for the s3 file containing the workflow configuration.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The state of the workflow. State is “running” if any execution is running, otherwise reflects most recent execution state.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
    hour
```

time_zone : string

The time zone of this workflow.

next_execution_at : string/time

The time of the next scheduled execution.

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes
    successfully.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- failure_on : boolean
    If failure email notifications are on
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

created_at : string/time

updated_at : string/time

post (*name*, *, *from_job_chain*='DEFAULT', *definition*='DEFAULT', *schedule*='DEFAULT',
time_zone='DEFAULT', *notifications*='DEFAULT', *hidden*='DEFAULT')
Create a Workflow

Parameters *name* : string

The name of this workflow.

from_job_chain : integer, optional

If specified, create a workflow from the job chain this job is in, and inherit the schedule from the root of the chain.

definition : string, optional

The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

time_zone : string, optional

The time zone of this workflow.

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
```



```
- failure_on : boolean
    If failure email notifications are on
```

hidden : boolean, optional

The hidden status of the object.

Returns id : integer

The ID for this workflow.

name : string

The name of this workflow.

definition : string

The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.

valid : boolean

The validity of the workflow definition.

validation_errors : string

The errors encountered when validating the workflow definition.

file_id : string

The file id for the s3 file containing the workflow configuration.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The state of the workflow. State is “running” if any execution is running, otherwise reflects most recent execution state.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

time_zone : string

The time zone of this workflow.

next_execution_at : string/time

The time of the next scheduled execution.

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- failure_on : boolean
    If failure email notifications are on
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

created_at : string/time

updated_at : string/time

post_clone (*id*, *, *clone_schedule*='DEFAULT', *clone_notifications*='DEFAULT')

Clone this workflow

Parameters **id** : integer

The ID for the workflow.

clone_schedule : boolean, optional

If true, also copy the schedule to the new workflow.

clone_notifications : boolean, optional

If true, also copy the notifications to the new workflow.

Returns **id** : integer

The ID for this workflow.

name : string

The name of this workflow.

definition : string

The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.

valid : boolean

The validity of the workflow definition.

validation_errors : string

The errors encountered when validating the workflow definition.

file_id : string

The file id for the s3 file containing the workflow configuration.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The state of the workflow. State is “running” if any execution is running, otherwise reflects most recent execution state.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

time_zone : string

The time zone of this workflow.

next_execution_at : string/time

The time of the next scheduled execution.

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- failure_email_addresses : list
```

```
Addresses to notify by e-mail when the job fails.
- success_on : boolean
  If success email notifications are on
- failure_on : boolean
  If failure email notifications are on
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

created_at : string/time

updated_at : string/time

post_executions (*id*, *, *target_task*='DEFAULT')

Execute a workflow

Parameters *id* : integer

The ID for the workflow.

target_task : string, optional

For a reverse workflow, the name of the task to target.

Returns *id* : integer

The ID for this workflow execution.

state : string

The state of this workflow execution.

mistral_state : string

The state of this workflow as reported by mistral. One of running, paused, success, error, or cancelled

mistral_state_info : string

The state info of this workflow as reported by mistral.

user : dict:

```
- id : integer
  The ID of this user.
- name : string
  This user's name.
- username : string
  This user's username.
- initials : string
  This user's initials.
- online : boolean
  Whether this user is online.
```

definition : string

The definition of the workflow for this execution.

tasks : list:

```

The tasks associated with this execution.
- name : string
    The name of the task.
- mistral_state : string
    The state of this task. One of idle, waiting, running, running_
    ↪delayed,
    success, or error
- mistral_state_info : string
    Extra info associated with the state of the task.
- runs : list::
    The runs associated with this task, in descending order by id.
    - id : integer
        The ID of the run.
    - job_id : integer
        The ID of the job associated with the run.

```

started_at : string/time

The time this execution started.

finished_at : string/time

The time this execution finished.

created_at : string/time

The time this execution was created.

updated_at : string/time

The time this execution was last updated.

post_executions_cancel (*id*, *execution_id*)

Cancel a workflow execution

Parameters **id** : integer

The ID for the workflow.

execution_id : integer

The ID for the workflow execution.

Returns **id** : integer

The ID for this workflow execution.

state : string

The state of this workflow execution.

mistral_state : string

The state of this workflow as reported by mistral. One of running, paused, success, error, or cancelled

mistral_state_info : string

The state info of this workflow as reported by mistral.

user : dict:

```

- id : integer
    The ID of this user.
- name : string

```

```
This user's name.
- username : string
  This user's username.
- initials : string
  This user's initials.
- online : boolean
  Whether this user is online.
```

definition : string

The definition of the workflow for this execution.

tasks : list:

```
The tasks associated with this execution.
- name : string
  The name of the task.
- mistral_state : string
  The state of this task. One of idle, waiting, running, running_
  ↳ delayed,
  success, or error
- mistral_state_info : string
  Extra info associated with the state of the task.
- runs : list::
  The runs associated with this task, in descending order by id.
  - id : integer
    The ID of the run.
  - job_id : integer
    The ID of the job associated with the run.
```

started_at : string/time

The time this execution started.

finished_at : string/time

The time this execution finished.

created_at : string/time

The time this execution was created.

updated_at : string/time

The time this execution was last updated.

post_executions_resume (*id*, *execution_id*)

Resume a paused workflow execution

Parameters *id* : integer

The ID for the workflow.

execution_id : integer

The ID for the workflow execution.

Returns *id* : integer

The ID for this workflow execution.

state : string

The state of this workflow execution.

mistral_state : string

The state of this workflow as reported by mistral. One of running, paused, success, error, or cancelled

mistral_state_info : string

The state info of this workflow as reported by mistral.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

definition : string

The definition of the workflow for this execution.

tasks : list:

```
The tasks associated with this execution.
- name : string
    The name of the task.
- mistral_state : string
    The state of this task. One of idle, waiting, running, running_
    delayed,
    success, or error
- mistral_state_info : string
    Extra info associated with the state of the task.
- runs : list::
    The runs associated with this task, in descending order by id.
    - id : integer
        The ID of the run.
    - job_id : integer
        The ID of the job associated with the run.
```

started_at : string/time

The time this execution started.

finished_at : string/time

The time this execution finished.

created_at : string/time

The time this execution was created.

updated_at : string/time

The time this execution was last updated.

post_executions_retry (*id*, *execution_id*, *, *task_name*='DEFAULT')

Retry a failed task, or all failed tasks in an execution

Parameters *id* : integer

The ID for the workflow.

execution_id : integer

The ID for the workflow execution.

task_name : string, optional

If specified, the name of the task to be retried. If not specified, all failed tasks in the execution will be retried.

Returns **id** : integer

The ID for this workflow execution.

state : string

The state of this workflow execution.

mistral_state : string

The state of this workflow as reported by mistral. One of running, paused, success, error, or cancelled

mistral_state_info : string

The state info of this workflow as reported by mistral.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

definition : string

The definition of the workflow for this execution.

tasks : list:

```
The tasks associated with this execution.
- name : string
    The name of the task.
- mistral_state : string
    The state of this task. One of idle, waiting, running, running_
↳ delayed,
    success, or error
- mistral_state_info : string
    Extra info associated with the state of the task.
- runs : list::
    The runs associated with this task, in descending order by id.
    - id : integer
        The ID of the run.
    - job_id : integer
        The ID of the job associated with the run.
```

started_at : string/time

The time this execution started.

finished_at : string/time

The time this execution finished.

created_at : string/time

The time this execution was created.

updated_at : string/time

The time this execution was last updated.

put (*id*, *name*, *, *definition*='DEFAULT', *schedule*='DEFAULT', *time_zone*='DEFAULT', *notifications*='DEFAULT')

Replace all attributes of this Workflow

Parameters *id* : integer

The ID for this workflow.

name : string

The name of this workflow.

definition : string, optional

The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

time_zone : string, optional

The time zone of this workflow.

notifications : dict, optional:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
```

```
- failure_on : boolean
    If failure email notifications are on
```

Returns `id` : integer

The ID for this workflow.

name : string

The name of this workflow.

definition : string

The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.

valid : boolean

The validity of the workflow definition.

validation_errors : string

The errors encountered when validating the workflow definition.

file_id : string

The file id for the s3 file containing the workflow configuration.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The state of the workflow. State is “running” if any execution is running, otherwise reflects most recent execution state.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

time_zone : string

The time zone of this workflow.

next_execution_at : string/time

The time of the next scheduled execution.

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- failure_on : boolean
    If failure email notifications are on
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

created_at : string/time

updated_at : string/time

put_archive (*id*, *status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **id** : integer

The ID for this workflow.

name : string

The name of this workflow.

definition : string

The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.

valid : boolean

The validity of the workflow definition.

validation_errors : string

The errors encountered when validating the workflow definition.

file_id : string

The file id for the s3 file containing the workflow configuration.

user : dict:

```
- id : integer
    The ID of this user.
- name : string
    This user's name.
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
    Whether this user is online.
```

state : string

The state of the workflow. State is “running” if any execution is running, otherwise reflects most recent execution state.

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
```

time_zone : string

The time zone of this workflow.

next_execution_at : string/time

The time of the next scheduled execution.

notifications : dict:

```
- urls : list
    URLs to receive a POST request at job completion
- success_email_subject : string
    Custom subject line for success e-mail.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- failure_on : boolean
    If failure email notifications are on
```

archived : string

The archival status of the requested object(s).

hidden : boolean

The hidden status of the object.

created_at : string/time

updated_at : string/time

put_projects (*id*, *project_id*)

Add a Workflow::Workflow to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_shares_groups (*id*, *group_ids*, *permission_level*, *, *share_email_body*=*'DEFAULT'*,
send_shared_email=*'DEFAULT'*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_shares_users (*id*, *user_ids*, *permission_level*, *, *share_email_body*='DEFAULT', *send_shared_email*='DEFAULT')

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: “read”, “write”, or “manage”

share_email_body : string, optional

Custom body text for e-mail sent on a share.

send_shared_email : boolean, optional

Send email to the recipients of a share.

Returns **readers** : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

writers : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

owners : dict:

```
- users : list::
  - id : integer
  - name : string
- groups : list::
  - id : integer
  - name : string
```

total_user_shares : integer

For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

5.6 Command Line Interface

A command line interface (CLI) to Civis is provided. This can be invoked by typing the command `civis` in the shell (sh, bash, zsh, etc.). It can also be used in Civis container scripts where the Docker image has this client installed. Here's a simple example of printing the types of scripts.

```
> civis scripts list-types
- name: sql
- name: python3
- name: javascript
- name: r
- name: containers
```

Not all API endpoints are available through the CLI since some take complex data types (e.g., arrays, objects/dictionaries) as input. However, functionality is available for getting information about scripts, logs, etc., as well as executing already created scripts.

There are a few extra, CLI-only commands that wrap the Files API endpoints to make uploading and downloading files easier: `civis files upload $PATH` and `civis files download $FILEID $PATH`.

The default output format is YAML, but the `--json-output` allows you to get output in JSON.

5.6.1 Notebooks

The following CLI-only commands make it easier to use Civis Platform as a backend for your Jupyter notebooks.

- `civis notebooks download $NOTEBOOK_ID $PATH`

Download a notebook from Civis Platform to the requested file on the local filesystem.

- `civis notebooks new [$LANGUAGE] [--mem $MEMORY] [--cpu $CPU]`

Create a new notebook, allocate resources for it, and open it in a tab of your default web browser. This command is the most similar to `jupyter notebook`. By default, Civis Platform will create a Python 3 notebook, but you can request any other language. Optional resource parameters let you allocate more memory or CPU to your notebook.

- `civis notebooks up $NOTEBOOK_ID [--mem $MEMORY] [--cpu $CPU]`

Allocate resources for a notebook which already exists in Civis Platform and open it in a tab of your default browser. Optional resource arguments allow you to change resources allocated to your notebook (default to using the same resources as the previous run).

- `civis notebooks down $NOTEBOOK_ID`

Stop a running notebook and free up the resources allocated to it.

- `civis notebooks open $NOTEBOOK_ID`

Open an existing notebook (which may or may not be running) in your default browser.

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

C

`civis.parallel`, 46

A

add_done_callback() (civis.ml.ModelFuture method), 41
 Announcements (class in civis.resources._resources), 56
 APIClient (class in civis), 51
 Apps (class in civis.resources._resources), 57

C

cancel() (civis.ml.ModelFuture method), 41
 cancelled() (civis.ml.ModelFuture method), 41
 civis.parallel (module), 46
 CIVIS_API_KEY, 14, 16, 17, 19–28, 35, 37, 40, 51, 56
 civis_to_csv() (in module civis.io), 14
 civis_to_file() (in module civis.io), 23
 civis_to_multifile_csv() (in module civis.io), 15
 CivisFuture (class in civis.futures), 55
 Clusters (class in civis.resources._resources), 68
 Codes (class in civis.resources._resources), 74
 Credentials (class in civis.resources._resources), 78
 csv_to_civis() (in module civis.io), 17

D

Databases (class in civis.resources._resources), 86
 dataframe_to_civis() (in module civis.io), 19
 default_credential (civis.APIClient attribute), 53
 delete() (civis.resources._resources.Codes method), 74
 delete_api_keys() (civis.resources._resources.Users method), 638
 delete_builds() (civis.resources._resources.Models method), 206
 delete_cass_ncoa_projects() (civis.resources._resources.Enhancements method), 90
 delete_cass_ncoa_runs() (civis.resources._resources.Enhancements method), 90
 delete_cass_ncoa_shares_groups() (civis.resources._resources.Enhancements method), 90
 delete_cass_ncoa_shares_users() (civis.resources._resources.Enhancements method), 90

method), 90
 delete_containers_projects() (civis.resources._resources.Scripts method), 354
 delete_containers_runs() (civis.resources._resources.Scripts method), 354
 delete_containers_shares_groups() (civis.resources._resources.Scripts method), 354
 delete_containers_shares_users() (civis.resources._resources.Scripts method), 355
 delete_custom_projects() (civis.resources._resources.Scripts method), 355
 delete_custom_runs() (civis.resources._resources.Scripts method), 355
 delete_custom_shares_groups() (civis.resources._resources.Scripts method), 355
 delete_custom_shares_users() (civis.resources._resources.Scripts method), 356
 delete_deployments() (civis.resources._resources.Notebooks method), 237
 delete_files_runs() (civis.resources._resources.Imports method), 129
 delete_grants() (civis.resources._resources.Reports method), 299
 delete_grants() (civis.resources._resources.Results method), 325
 delete_instances_projects() (civis.resources._resources.Apps method), 58
 delete_instances_shares_groups() (civis.resources._resources.Apps method), 58
 delete_instances_shares_users() (civis.resources._resources.Apps method), 58

`delete_javascript_projects()`
(`civis.resources._resources.Scripts` method), 356

`delete_javascript_runs()` (`civis.resources._resources.Scripts` method), 356

`delete_javascript_shares_groups()`
(`civis.resources._resources.Scripts` method), 356

`delete_javascript_shares_users()`
(`civis.resources._resources.Scripts` method), 356

`delete_optimizations_runs()`
(`civis.resources._resources.Media` method), 184

`delete_optimizations_shares_groups()`
(`civis.resources._resources.Media` method), 184

`delete_optimizations_shares_users()`
(`civis.resources._resources.Media` method), 185

`delete_projects()` (`civis.resources._resources.Files` method), 121

`delete_projects()` (`civis.resources._resources.Imports` method), 129

`delete_projects()` (`civis.resources._resources.Jobs` method), 174

`delete_projects()` (`civis.resources._resources.Models` method), 206

`delete_projects()` (`civis.resources._resources.Notebooks` method), 237

`delete_projects()` (`civis.resources._resources.Reports` method), 299

`delete_projects()` (`civis.resources._resources.Results` method), 325

`delete_projects()` (`civis.resources._resources.Workflows` method), 650

`delete_python3_projects()`
(`civis.resources._resources.Scripts` method), 357

`delete_python3_runs()` (`civis.resources._resources.Scripts` method), 357

`delete_python3_shares_groups()`
(`civis.resources._resources.Scripts` method), 357

`delete_python3_shares_users()`
(`civis.resources._resources.Scripts` method), 357

`delete_r_projects()` (`civis.resources._resources.Scripts` method), 357

`delete_r_runs()` (`civis.resources._resources.Scripts` method), 358

`delete_r_shares_groups()`
(`civis.resources._resources.Scripts` method), 358

`delete_r_shares_users()` (`civis.resources._resources.Scripts` method), 358

`delete_reports_shares_groups()`
(`civis.resources._resources.Templates` method), 619

`delete_reports_shares_users()`
(`civis.resources._resources.Templates` method), 619

`delete_runs()` (`civis.resources._resources.Predictions` method), 263

`delete_runs()` (`civis.resources._resources.Queries` method), 289

`delete_scripts_projects()` (`civis.resources._resources.Templates` method), 620

`delete_scripts_shares_groups()`
(`civis.resources._resources.Templates` method), 620

`delete_scripts_shares_users()`
(`civis.resources._resources.Templates` method), 620

`delete_shares_groups()` (`civis.resources._resources.Credentials` method), 78

`delete_shares_groups()` (`civis.resources._resources.Files` method), 121

`delete_shares_groups()` (`civis.resources._resources.Imports` method), 130

`delete_shares_groups()` (`civis.resources._resources.Jobs` method), 174

`delete_shares_groups()` (`civis.resources._resources.Models` method), 206

`delete_shares_groups()` (`civis.resources._resources.Notebooks` method), 237

`delete_shares_groups()` (`civis.resources._resources.Projects` method), 272

`delete_shares_groups()` (`civis.resources._resources.Reports` method), 299

`delete_shares_groups()` (`civis.resources._resources.Results` method), 325

`delete_shares_groups()` (`civis.resources._resources.Workflows` method), 650

`delete_shares_users()` (`civis.resources._resources.Credentials` method), 78

`delete_shares_users()` (`civis.resources._resources.Files` method), 122

`delete_shares_users()` (`civis.resources._resources.Imports` method), 130

`delete_shares_users()` (`civis.resources._resources.Jobs` method), 174

`delete_shares_users()` (`civis.resources._resources.Models` method), 206

`delete_shares_users()` (`civis.resources._resources.Notebooks` method), 237

`delete_shares_users()` (`civis.resources._resources.Projects` method), 272

- delete_shares_users() (civis.resources._resources.Reports method), 299
- delete_shares_users() (civis.resources._resources.Results method), 326
- delete_shares_users() (civis.resources._resources.Workflows method), 650
- delete_solutions_shares_groups()
(civis.resources._resources.Reports method), 300
- delete_solutions_shares_groups()
(civis.resources._resources.Results method), 326
- delete_solutions_shares_users()
(civis.resources._resources.Reports method), 300
- delete_solutions_shares_users()
(civis.resources._resources.Results method), 326
- delete_spot_orders_shares_groups()
(civis.resources._resources.Media method), 185
- delete_spot_orders_shares_users()
(civis.resources._resources.Media method), 185
- delete_sql_projects() (civis.resources._resources.Scripts method), 358
- delete_sql_runs() (civis.resources._resources.Scripts method), 358
- delete_sql_shares_groups()
(civis.resources._resources.Scripts method), 359
- delete_sql_shares_users()
(civis.resources._resources.Scripts method), 359
- delete_whitelist_ips() (civis.resources._resources.Databases method), 86
- done() (civis.ml.ModelFuture method), 41
- ## E
- Endpoints (class in civis.resources._resources), 89
- Enhancements (class in civis.resources._resources), 89
- environment variable
CIVIS_API_KEY, 14, 16, 17, 19–28, 35, 37, 40, 51, 56
- exception() (civis.ml.ModelFuture method), 42
- Exports (class in civis.resources._resources), 119
- ## F
- failed() (civis.ml.ModelFuture method), 42
- file_id_from_run_output() (in module civis.io), 24
- file_to_civis() (in module civis.io), 25
- file_to_dataframe() (in module civis.io), 25
- file_to_json() (in module civis.io), 26
- Files (class in civis.resources._resources), 121
- from_existing() (civis.ml.ModelPipeline class method), 36
- ## G
- get() (civis.resources._resources.Apps method), 59
- get() (civis.resources._resources.Codes method), 75
- get() (civis.resources._resources.Credentials method), 78
- get() (civis.resources._resources.Files method), 122
- get() (civis.resources._resources.Imports method), 130
- get() (civis.resources._resources.Jobs method), 175
- get() (civis.resources._resources.Models method), 207
- get() (civis.resources._resources.Notebooks method), 238
- get() (civis.resources._resources.Predictions method), 264
- get() (civis.resources._resources.Projects method), 273
- get() (civis.resources._resources.Queries method), 289
- get() (civis.resources._resources.Reports method), 300
- get() (civis.resources._resources.Results method), 326
- get() (civis.resources._resources.Scripts method), 359
- get() (civis.resources._resources.Tables method), 601
- get() (civis.resources._resources.Users method), 639
- get() (civis.resources._resources.Workflows method), 650
- get_api_keys() (civis.resources._resources.Users method), 640
- get_aws_credential_id() (civis.APIClient method), 53
- get_batches() (civis.resources._resources.Imports method), 134
- get_builds() (civis.resources._resources.Models method), 210
- get_cass_ncoa() (civis.resources._resources.Enhancements method), 90
- get_cass_ncoa_runs() (civis.resources._resources.Enhancements method), 93
- get_containers() (civis.resources._resources.Scripts method), 363
- get_containers_runs() (civis.resources._resources.Scripts method), 367
- get_custom() (civis.resources._resources.Scripts method), 367
- get_custom_runs() (civis.resources._resources.Scripts method), 371
- get_database_credential_id() (civis.APIClient method), 53
- get_database_id() (civis.APIClient method), 54
- get_deployments() (civis.resources._resources.Notebooks method), 240
- get_enhancements_cass_ncoa()
(civis.resources._resources.Tables method), 604
- get_enhancements_geocodings()
(civis.resources._resources.Tables method), 605
- get_enhancements_prepared_matchings()
(civis.resources._resources.Tables method),

605
 get_enhancements_table_matchings()
 (civis.resources._resources.Tables method),
 606
 get_executions() (civis.resources._resources.Workflows
 method), 652
 get_files_runs() (civis.resources._resources.Imports
 method), 135
 get_instances() (civis.resources._resources.Apps
 method), 59
 get_javascript() (civis.resources._resources.Scripts
 method), 371
 get_javascript_runs() (civis.resources._resources.Scripts
 method), 375
 get_kubernetes() (civis.resources._resources.Clusters
 method), 68
 get_optimizations() (civis.resources._resources.Media
 method), 185
 get_optimizations_runs()
 (civis.resources._resources.Media method),
 187
 get_python3() (civis.resources._resources.Scripts
 method), 375
 get_python3_runs() (civis.resources._resources.Scripts
 method), 379
 get_r() (civis.resources._resources.Scripts method), 380
 get_r_runs() (civis.resources._resources.Scripts method),
 384
 get_reports() (civis.resources._resources.Templates
 method), 620
 get_runs() (civis.resources._resources.Jobs method), 176
 get_runs() (civis.resources._resources.Predictions
 method), 265
 get_runs() (civis.resources._resources.Queries method),
 291
 get_scripts() (civis.resources._resources.Templates
 method), 621
 get_solutions() (civis.resources._resources.Reports
 method), 302
 get_solutions() (civis.resources._resources.Results
 method), 328
 get_spot_orders() (civis.resources._resources.Media
 method), 187
 get_sql() (civis.resources._resources.Scripts method),
 384
 get_sql_runs() (civis.resources._resources.Scripts
 method), 388
 get_table_id() (civis.APIClient method), 54
 get_whitelist_ips() (civis.resources._resources.Databases
 method), 87
 get_workers() (civis.resources._resources.Clusters
 method), 69
 Groups (class in civis.resources._resources), 128

I

Imports (class in civis.resources._resources), 129
 infer_backend_factory() (in module civis.parallel), 46

J

Jobs (class in civis.resources._resources), 174

L

list() (civis.resources._resources.Announcements
 method), 57
 list() (civis.resources._resources.Apps method), 60
 list() (civis.resources._resources.Codes method), 75
 list() (civis.resources._resources.Credentials method), 79
 list() (civis.resources._resources.Databases method), 87
 list() (civis.resources._resources.Endpoints method), 89
 list() (civis.resources._resources.Enhancements method),
 94
 list() (civis.resources._resources.Exports method), 119
 list() (civis.resources._resources.Groups method), 128
 list() (civis.resources._resources.Imports method), 136
 list() (civis.resources._resources.Jobs method), 176
 list() (civis.resources._resources.Models method), 211
 list() (civis.resources._resources.Notebooks method), 241
 list() (civis.resources._resources.Notifications method),
 262
 list() (civis.resources._resources.Ontology method), 263
 list() (civis.resources._resources.Predictions method),
 266
 list() (civis.resources._resources.Projects method), 276
 list() (civis.resources._resources.Queries method), 291
 list() (civis.resources._resources.Reports method), 303
 list() (civis.resources._resources.Results method), 329
 list() (civis.resources._resources.Scripts method), 389
 list() (civis.resources._resources.Search method), 599
 list() (civis.resources._resources.Tables method), 607
 list() (civis.resources._resources.Users method), 641
 list() (civis.resources._resources.Workflows method), 653
 list_api_keys() (civis.resources._resources.Users
 method), 642
 list_batches() (civis.resources._resources.Imports
 method), 138
 list_builds() (civis.resources._resources.Models method),
 215
 list_builds_logs() (civis.resources._resources.Models
 method), 216
 list_cass_ncoa_projects()
 (civis.resources._resources.Enhancements
 method), 95
 list_cass_ncoa_runs() (civis.resources._resources.Enhancements
 method), 96
 list_cass_ncoa_runs_logs()
 (civis.resources._resources.Enhancements
 method), 97

[list_cass_ncoa_runs_outputs\(\)](#)
 (civis.resources._resources.Enhancements method), 97

[list_cass_ncoa_shares\(\)](#) (civis.resources._resources.Enhancements method), 98

[list_children\(\)](#) (civis.resources._resources.Jobs method), 177

[list_columns\(\)](#) (civis.resources._resources.Tables method), 609

[list_containers_projects\(\)](#)
 (civis.resources._resources.Scripts method), 391

[list_containers_runs\(\)](#) (civis.resources._resources.Scripts method), 392

[list_containers_runs_logs\(\)](#)
 (civis.resources._resources.Scripts method), 393

[list_containers_runs_outputs\(\)](#)
 (civis.resources._resources.Scripts method), 394

[list_containers_shares\(\)](#) (civis.resources._resources.Scripts method), 395

[list_custom\(\)](#) (civis.resources._resources.Scripts method), 395

[list_custom_projects\(\)](#) (civis.resources._resources.Scripts method), 397

[list_custom_runs\(\)](#) (civis.resources._resources.Scripts method), 398

[list_custom_runs_logs\(\)](#) (civis.resources._resources.Scripts method), 399

[list_custom_runs_outputs\(\)](#)
 (civis.resources._resources.Scripts method), 400

[list_custom_shares\(\)](#) (civis.resources._resources.Scripts method), 400

[list_deployments\(\)](#) (civis.resources._resources.Notebooks method), 242

[list_dmas\(\)](#) (civis.resources._resources.Media method), 188

[list_executions\(\)](#) (civis.resources._resources.Workflows method), 655

[list_files_runs\(\)](#) (civis.resources._resources.Imports method), 139

[list_files_runs_logs\(\)](#) (civis.resources._resources.Imports method), 140

[list_history\(\)](#) (civis.resources._resources.Scripts method), 401

[list_instances\(\)](#) (civis.resources._resources.Apps method), 60

[list_instances_projects\(\)](#) (civis.resources._resources.Apps method), 61

[list_instances_shares\(\)](#) (civis.resources._resources.Apps method), 62

[list_javascript_projects\(\)](#) (civis.resources._resources.Scripts method), 402

[list_javascript_runs\(\)](#) (civis.resources._resources.Scripts method), 403

[list_javascript_runs_logs\(\)](#)
 (civis.resources._resources.Scripts method), 403

[list_javascript_runs_outputs\(\)](#)
 (civis.resources._resources.Scripts method), 404

[list_javascript_shares\(\)](#) (civis.resources._resources.Scripts method), 405

[list_kubernetes\(\)](#) (civis.resources._resources.Clusters method), 70

[list_me\(\)](#) (civis.resources._resources.Users method), 643

[list_optimizations\(\)](#) (civis.resources._resources.Media method), 188

[list_optimizations_runs\(\)](#)
 (civis.resources._resources.Media method), 189

[list_optimizations_runs_logs\(\)](#)
 (civis.resources._resources.Media method), 190

[list_optimizations_shares\(\)](#)
 (civis.resources._resources.Media method), 190

[list_parents\(\)](#) (civis.resources._resources.Jobs method), 178

[list_projects\(\)](#) (civis.resources._resources.Files method), 122

[list_projects\(\)](#) (civis.resources._resources.Imports method), 140

[list_projects\(\)](#) (civis.resources._resources.Jobs method), 179

[list_projects\(\)](#) (civis.resources._resources.Models method), 217

[list_projects\(\)](#) (civis.resources._resources.Notebooks method), 244

[list_projects\(\)](#) (civis.resources._resources.Reports method), 305

[list_projects\(\)](#) (civis.resources._resources.Results method), 331

[list_projects\(\)](#) (civis.resources._resources.Workflows method), 656

[list_python3_projects\(\)](#) (civis.resources._resources.Scripts method), 405

[list_python3_runs\(\)](#) (civis.resources._resources.Scripts method), 406

[list_python3_runs_logs\(\)](#)
 (civis.resources._resources.Scripts method), 407

[list_python3_runs_outputs\(\)](#)
 (civis.resources._resources.Scripts method), 408

[list_python3_shares\(\)](#) (civis.resources._resources.Scripts method), 408

method), 409

list_r_projects() (civis.resources._resources.Scripts method), 409

list_r_runs() (civis.resources._resources.Scripts method), 410

list_r_runs_logs() (civis.resources._resources.Scripts method), 411

list_r_runs_outputs() (civis.resources._resources.Scripts method), 412

list_r_shares() (civis.resources._resources.Scripts method), 412

list_ratecards() (civis.resources._resources.Media method), 191

list_reports() (civis.resources._resources.Templates method), 622

list_reports_shares() (civis.resources._resources.Templates method), 623

list_runs() (civis.resources._resources.Imports method), 141

list_runs() (civis.resources._resources.Predictions method), 267

list_runs() (civis.resources._resources.Queries method), 293

list_runs_logs() (civis.resources._resources.Imports method), 142

list_runs_logs() (civis.resources._resources.Predictions method), 268

list_runs_logs() (civis.resources._resources.Queries method), 294

list_schedules() (civis.resources._resources.Models method), 218

list_schedules() (civis.resources._resources.Predictions method), 268

list_schemas() (civis.resources._resources.Databases method), 87

list_scripts() (civis.resources._resources.Templates method), 624

list_scripts_projects() (civis.resources._resources.Templates method), 625

list_scripts_shares() (civis.resources._resources.Templates method), 626

list_shares() (civis.resources._resources.Credentials method), 80

list_shares() (civis.resources._resources.Files method), 123

list_shares() (civis.resources._resources.Imports method), 142

list_shares() (civis.resources._resources.Jobs method), 180

list_shares() (civis.resources._resources.Models method), 218

list_shares() (civis.resources._resources.Notebooks method), 244

list_shares() (civis.resources._resources.Projects method), 277

list_shares() (civis.resources._resources.Reports method), 306

list_shares() (civis.resources._resources.Results method), 332

list_shares() (civis.resources._resources.Workflows method), 657

list_snapshots() (civis.resources._resources.Reports method), 306

list_snapshots() (civis.resources._resources.Results method), 333

list_solutions_shares() (civis.resources._resources.Reports method), 307

list_solutions_shares() (civis.resources._resources.Results method), 334

list_spot_orders() (civis.resources._resources.Media method), 192

list_spot_orders_shares() (civis.resources._resources.Media method), 192

list_sql_projects() (civis.resources._resources.Scripts method), 413

list_sql_runs() (civis.resources._resources.Scripts method), 414

list_sql_runs_logs() (civis.resources._resources.Scripts method), 415

list_sql_runs_outputs() (civis.resources._resources.Scripts method), 416

list_sql_shares() (civis.resources._resources.Scripts method), 416

list_targets() (civis.resources._resources.Media method), 193

list_types() (civis.resources._resources.Enhancements method), 99

list_types() (civis.resources._resources.Models method), 219

list_types() (civis.resources._resources.Scripts method), 417

list_types() (civis.resources._resources.Search method), 600

list_update_links() (civis.resources._resources.Notebooks method), 245

list_whitelist_ips() (civis.resources._resources.Databases method), 88

list_workers() (civis.resources._resources.Clusters method), 71

list_workers_active_jobs() (civis.resources._resources.Clusters method), 71

list_workers_queued_jobs() (civis.resources._resources.Clusters method), 73

M

`make_backend_factory()` (in module `civis.parallel`), 48
`make_backend_template_factory()` (in module `civis.parallel`), 50

`Match_Targets` (in module `civis.resources._resources`), 183

`Media` (class in `civis.resources._resources`), 183

`ModelFuture` (class in `civis.ml`), 40

`ModelPipeline` (class in `civis.ml`), 34

`Models` (class in `civis.resources._resources`), 205

N

`Notebooks` (class in `civis.resources._resources`), 236

`Notifications` (class in `civis.resources._resources`), 262

O

`Ontology` (class in `civis.resources._resources`), 263

P

`PaginatedResponse` (class in `civis.response`), 55

`patch()` (`civis.resources._resources.Codes` method), 76

`patch()` (`civis.resources._resources.Models` method), 219

`patch()` (`civis.resources._resources.Notebooks` method), 245

`patch()` (`civis.resources._resources.Predictions` method), 269

`patch()` (`civis.resources._resources.Reports` method), 308

`patch()` (`civis.resources._resources.Results` method), 334

`patch()` (`civis.resources._resources.Scripts` method), 417

`patch()` (`civis.resources._resources.Tables` method), 610

`patch()` (`civis.resources._resources.Workflows` method), 657

`patch_cass_ncoa()` (`civis.resources._resources.Enhancements` method), 99

`patch_containers()` (`civis.resources._resources.Scripts` method), 422

`patch_custom()` (`civis.resources._resources.Scripts` method), 429

`patch_instances()` (`civis.resources._resources.Apps` method), 63

`patch_javascript()` (`civis.resources._resources.Scripts` method), 434

`patch_me()` (`civis.resources._resources.Users` method), 644

`patch_optimizations()` (`civis.resources._resources.Media` method), 193

`patch_python3()` (`civis.resources._resources.Scripts` method), 439

`patch_r()` (`civis.resources._resources.Scripts` method), 445

`patch_reports()` (`civis.resources._resources.Templates` method), 626

`patch_scripts()` (`civis.resources._resources.Templates` method), 628

`patch_snapshots()` (`civis.resources._resources.Reports` method), 310

`patch_snapshots()` (`civis.resources._resources.Results` method), 337

`patch_sql()` (`civis.resources._resources.Scripts` method), 451

`post()` (`civis.resources._resources.Codes` method), 76

`post()` (`civis.resources._resources.Credentials` method), 81

`post()` (`civis.resources._resources.Files` method), 124

`post()` (`civis.resources._resources.Imports` method), 143

`post()` (`civis.resources._resources.Models` method), 221

`post()` (`civis.resources._resources.Notebooks` method), 248

`post()` (`civis.resources._resources.Projects` method), 278

`post()` (`civis.resources._resources.Queries` method), 294

`post()` (`civis.resources._resources.Reports` method), 312

`post()` (`civis.resources._resources.Results` method), 339

`post()` (`civis.resources._resources.Scripts` method), 458

`post()` (`civis.resources._resources.Tables` method), 611

`post()` (`civis.resources._resources.Workflows` method), 660

`post_api_keys()` (`civis.resources._resources.Users` method), 648

`post_authenticate()` (`civis.resources._resources.Credentials` method), 82

`post_batches()` (`civis.resources._resources.Imports` method), 149

`post_builds()` (`civis.resources._resources.Models` method), 227

`post_cancel()` (`civis.resources._resources.Imports` method), 150

`post_cancel()` (`civis.resources._resources.Scripts` method), 462

`post_cass_ncoa()` (`civis.resources._resources.Enhancements` method), 104

`post_cass_ncoa_cancel()` (`civis.resources._resources.Enhancements` method), 108

`post_cass_ncoa_runs()` (`civis.resources._resources.Enhancements` method), 109

`post_clone()` (`civis.resources._resources.Notebooks` method), 252

`post_clone()` (`civis.resources._resources.Workflows` method), 662

`post_containers()` (`civis.resources._resources.Scripts` method), 463

`post_containers_clone()` (`civis.resources._resources.Scripts` method), 469

`post_containers_runs()` (`civis.resources._resources.Scripts` method), 474

`post_containers_runs_outputs()` (`civis.resources._resources.Scripts` method), 474

`post_custom()` (`civis.resources._resources.Scripts` method), 474

method), 475
post_custom_clone() (civis.resources._resources.Scripts method), 479
post_custom_runs() (civis.resources._resources.Scripts method), 483
post_custom_runs_outputs() (civis.resources._resources.Scripts method), 483
post_deployments() (civis.resources._resources.Notebooks method), 254
post_enhancements_cass_ncoa() (civis.resources._resources.Tables method), 612
post_enhancements_geocodings() (civis.resources._resources.Tables method), 613
post_enhancements_prepared_matchings() (civis.resources._resources.Tables method), 613
post_enhancements_table_matchings() (civis.resources._resources.Tables method), 614
post_executions() (civis.resources._resources.Workflows method), 664
post_executions_cancel() (civis.resources._resources.Workflows method), 665
post_executions_resume() (civis.resources._resources.Workflows method), 666
post_executions_retry() (civis.resources._resources.Workflows method), 667
post_files() (civis.resources._resources.Imports method), 150
post_files_runs() (civis.resources._resources.Imports method), 151
post_grants() (civis.resources._resources.Reports method), 315
post_grants() (civis.resources._resources.Results method), 341
post_instances() (civis.resources._resources.Apps method), 64
post_javascript() (civis.resources._resources.Scripts method), 484
post_javascript_clone() (civis.resources._resources.Scripts method), 489
post_javascript_runs() (civis.resources._resources.Scripts method), 493
post_javascript_runs_outputs() (civis.resources._resources.Scripts method), 493
post_multipart() (civis.resources._resources.Files method), 125
post_multipart_complete() (civis.resources._resources.Files method), 125
post_optimizations() (civis.resources._resources.Media method), 195
post_optimizations_clone() (civis.resources._resources.Media method), 197
post_optimizations_runs() (civis.resources._resources.Media method), 198
post_python3() (civis.resources._resources.Scripts method), 494
post_python3_clone() (civis.resources._resources.Scripts method), 500
post_python3_runs() (civis.resources._resources.Scripts method), 504
post_python3_runs_outputs() (civis.resources._resources.Scripts method), 505
post_r() (civis.resources._resources.Scripts method), 505
post_r_clone() (civis.resources._resources.Scripts method), 511
post_r_runs() (civis.resources._resources.Scripts method), 515
post_r_runs_outputs() (civis.resources._resources.Scripts method), 516
post_refresh() (civis.resources._resources.Tables method), 615
post_reports() (civis.resources._resources.Templates method), 629
post_run() (civis.resources._resources.Scripts method), 516
post_runs() (civis.resources._resources.Imports method), 152
post_runs() (civis.resources._resources.Jobs method), 180
post_runs() (civis.resources._resources.Predictions method), 270
post_runs() (civis.resources._resources.Queries method), 296
post_scripts() (civis.resources._resources.Templates method), 630
post_snapshots() (civis.resources._resources.Reports method), 316
post_snapshots() (civis.resources._resources.Results method), 343
post_spot_orders() (civis.resources._resources.Media method), 198
post_sql() (civis.resources._resources.Scripts method), 516
post_sql_clone() (civis.resources._resources.Scripts method), 523
post_sql_runs() (civis.resources._resources.Scripts method), 527

- `post_syncs()` (civis.resources._resources.Imports method), 152
- `post_temporary()` (civis.resources._resources.Credentials method), 83
- `post_trigger_email()` (civis.resources._resources.Jobs method), 181
- `post_whitelist_ips()` (civis.resources._resources.Databases method), 88
- `predict()` (civis.ml.ModelPipeline method), 37
- `Predictions` (class in civis.resources._resources), 263
- `Projects` (class in civis.resources._resources), 272
- `put()` (civis.resources._resources.Codes method), 77
- `put()` (civis.resources._resources.Credentials method), 83
- `put()` (civis.resources._resources.Imports method), 155
- `put()` (civis.resources._resources.Notebooks method), 255
- `put()` (civis.resources._resources.Projects method), 281
- `put()` (civis.resources._resources.Workflows method), 669
- `put_archive()` (civis.resources._resources.Imports method), 161
- `put_archive()` (civis.resources._resources.Models method), 228
- `put_archive()` (civis.resources._resources.Notebooks method), 258
- `put_archive()` (civis.resources._resources.Projects method), 284
- `put_archive()` (civis.resources._resources.Reports method), 318
- `put_archive()` (civis.resources._resources.Results method), 345
- `put_archive()` (civis.resources._resources.Workflows method), 671
- `put_cass_ncoa()` (civis.resources._resources.Enhancements method), 109
- `put_cass_ncoa_archive()` (civis.resources._resources.Enhancements method), 114
- `put_cass_ncoa_projects()` (civis.resources._resources.Enhancements method), 117
- `put_cass_ncoa_shares_groups()` (civis.resources._resources.Enhancements method), 117
- `put_cass_ncoa_shares_users()` (civis.resources._resources.Enhancements method), 118
- `put_containers()` (civis.resources._resources.Scripts method), 528
- `put_containers_archive()` (civis.resources._resources.Scripts method), 534
- `put_containers_projects()` (civis.resources._resources.Scripts method), 539
- `put_containers_shares_groups()` (civis.resources._resources.Scripts method), 539
- `put_containers_shares_users()` (civis.resources._resources.Scripts method), 540
- `put_custom()` (civis.resources._resources.Scripts method), 541
- `put_custom_archive()` (civis.resources._resources.Scripts method), 545
- `put_custom_projects()` (civis.resources._resources.Scripts method), 549
- `put_custom_shares_groups()` (civis.resources._resources.Scripts method), 549
- `put_custom_shares_users()` (civis.resources._resources.Scripts method), 550
- `put_instances_archive()` (civis.resources._resources.Apps method), 65
- `put_instances_projects()` (civis.resources._resources.Apps method), 66
- `put_instances_shares_groups()` (civis.resources._resources.Apps method), 66
- `put_instances_shares_users()` (civis.resources._resources.Apps method), 67
- `put_javascript()` (civis.resources._resources.Scripts method), 551
- `put_javascript_archive()` (civis.resources._resources.Scripts method), 556
- `put_javascript_projects()` (civis.resources._resources.Scripts method), 560
- `put_javascript_shares_groups()` (civis.resources._resources.Scripts method), 560
- `put_javascript_shares_users()` (civis.resources._resources.Scripts method), 561
- `put_optimizations_archive()` (civis.resources._resources.Media method), 199
- `put_optimizations_shares_groups()` (civis.resources._resources.Media method), 200
- `put_optimizations_shares_users()` (civis.resources._resources.Media method), 201
- `put_predictions()` (civis.resources._resources.Models method), 232
- `put_projects()` (civis.resources._resources.Files method), 125
- `put_projects()` (civis.resources._resources.Imports method), 166

`put_projects()` (`civis.resources._resources.Jobs` method), 181

`put_projects()` (`civis.resources._resources.Models` method), 233

`put_projects()` (`civis.resources._resources.Notebooks` method), 260

`put_projects()` (`civis.resources._resources.Reports` method), 320

`put_projects()` (`civis.resources._resources.Results` method), 347

`put_projects()` (`civis.resources._resources.Workflows` method), 673

`put_python3()` (`civis.resources._resources.Scripts` method), 562

`put_python3_archive()` (`civis.resources._resources.Scripts` method), 568

`put_python3_projects()` (`civis.resources._resources.Scripts` method), 572

`put_python3_shares_groups()` (`civis.resources._resources.Scripts` method), 572

`put_python3_shares_users()` (`civis.resources._resources.Scripts` method), 573

`put_r()` (`civis.resources._resources.Scripts` method), 574

`put_r_archive()` (`civis.resources._resources.Scripts` method), 580

`put_r_projects()` (`civis.resources._resources.Scripts` method), 584

`put_r_shares_groups()` (`civis.resources._resources.Scripts` method), 584

`put_r_shares_users()` (`civis.resources._resources.Scripts` method), 585

`put_reports()` (`civis.resources._resources.Templates` method), 631

`put_reports_shares_groups()` (`civis.resources._resources.Templates` method), 632

`put_reports_shares_users()` (`civis.resources._resources.Templates` method), 633

`put_schedules()` (`civis.resources._resources.Models` method), 234

`put_schedules()` (`civis.resources._resources.Predictions` method), 271

`put_scripts()` (`civis.resources._resources.Queries` method), 297

`put_scripts()` (`civis.resources._resources.Templates` method), 634

`put_scripts_projects()` (`civis.resources._resources.Templates` method), 635

`put_scripts_shares_groups()` (`civis.resources._resources.Templates` method), 635

`put_scripts_shares_users()` (`civis.resources._resources.Templates` method), 636

`put_shares_groups()` (`civis.resources._resources.Credentials` method), 84

`put_shares_groups()` (`civis.resources._resources.Files` method), 126

`put_shares_groups()` (`civis.resources._resources.Imports` method), 166

`put_shares_groups()` (`civis.resources._resources.Jobs` method), 181

`put_shares_groups()` (`civis.resources._resources.Models` method), 234

`put_shares_groups()` (`civis.resources._resources.Notebooks` method), 260

`put_shares_groups()` (`civis.resources._resources.Projects` method), 287

`put_shares_groups()` (`civis.resources._resources.Reports` method), 320

`put_shares_groups()` (`civis.resources._resources.Results` method), 347

`put_shares_groups()` (`civis.resources._resources.Workflows` method), 673

`put_shares_users()` (`civis.resources._resources.Credentials` method), 85

`put_shares_users()` (`civis.resources._resources.Files` method), 127

`put_shares_users()` (`civis.resources._resources.Imports` method), 167

`put_shares_users()` (`civis.resources._resources.Jobs` method), 182

`put_shares_users()` (`civis.resources._resources.Models` method), 235

`put_shares_users()` (`civis.resources._resources.Notebooks` method), 261

`put_shares_users()` (`civis.resources._resources.Projects` method), 288

`put_shares_users()` (`civis.resources._resources.Reports` method), 321

`put_shares_users()` (`civis.resources._resources.Results` method), 348

`put_shares_users()` (`civis.resources._resources.Workflows` method), 674

`put_solutions_shares_groups()` (`civis.resources._resources.Reports` method), 322

`put_solutions_shares_groups()` (`civis.resources._resources.Results` method), 349

`put_solutions_shares_users()` (`civis.resources._resources.Reports` method), 323

`put_solutions_shares_users()` (`civis.resources._resources.Results` method), 349

350
 put_spot_orders() (civis.resources._resources.Media method), 202
 put_spot_orders_archive() (civis.resources._resources.Media method), 203
 put_spot_orders_shares_groups() (civis.resources._resources.Media method), 203
 put_spot_orders_shares_users() (civis.resources._resources.Media method), 204
 put_sql() (civis.resources._resources.Scripts method), 586
 put_sql_archive() (civis.resources._resources.Scripts method), 593
 put_sql_projects() (civis.resources._resources.Scripts method), 597
 put_sql_shares_groups() (civis.resources._resources.Scripts method), 597
 put_sql_shares_users() (civis.resources._resources.Scripts method), 598
 put_syncs() (civis.resources._resources.Imports method), 168
 put_syncs_archive() (civis.resources._resources.Imports method), 171

Q

Queries (class in civis.resources._resources), 289
 query_civis() (in module civis.io), 28

R

read_civis() (in module civis.io), 20
 read_civis_sql() (in module civis.io), 21
 Remote_Hosts (in module civis.resources._resources), 298
 Reports (class in civis.resources._resources), 298
 Response (class in civis.response), 55
 result() (civis.ml.ModelFuture method), 42
 Results (class in civis.resources._resources), 324
 running() (civis.ml.ModelFuture method), 42

S

Scripts (class in civis.resources._resources), 351
 Search (class in civis.resources._resources), 599
 set_exception() (civis.ml.ModelFuture method), 42
 set_result() (civis.ml.ModelFuture method), 42
 set_running_or_notify_cancel() (civis.ml.ModelFuture method), 42
 succeeded() (civis.ml.ModelFuture method), 43

T

Tables (class in civis.resources._resources), 600

Templates (class in civis.resources._resources), 618
 train() (civis.ml.ModelPipeline method), 38
 transfer_table() (in module civis.io), 27

U

username (civis.APIClient attribute), 54
 Users (class in civis.resources._resources), 637

W

Workflows (class in civis.resources._resources), 649