
Civis Client Documentation

Release 1.0.0

Civis Analytics

Jul 27, 2017

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	433
	Python Module Index	435

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. Install these dependencies with

```
pip install scikit-learn
pip install pandas
```


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*.

User Guide

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()
```

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 `fut` above forces the program to wait for the job started with `dataframe_to_civis()` to finish and returns the result.

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
```

Data Import and Export

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

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, cre-
                      dential_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.

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'`.

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: 'header', 'query', 'entries', representing the columns from the query, the query itself, and a list of dictionaries for each unloaded CSV part, each containing its file 'id', 'name', 'size', and unsigned and signed S3 urls, 'url' and 'url_signed', respectively.

See also:

`civis.APIClient.scripts.post_sql`

Examples

```
>>> sql = "SELECT * FROM schema.my_big_table"
>>> database = "my_database"
>>> delimiter = "|"
>>> manifest = civis.multipart_unload(sql, database, delimiter=delimiter)
>>> ids = [file['id'] for file in manifest['files']]
>>> 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', 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'.

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', 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'.

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.

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]
```

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

The file or other buffer to write the contents of the Civis file into.

`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
>>> with open("my_file.txt", "wb") as f:
...     civis_to_file(file_id, f)
```

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

The file or other buffer that you wish to upload.

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')`).

If you have the `requests-toolbelt` package installed (`pip install requests-toolbelt`) and the file-like object is seekable, then this function will stream from the open file pointer into Platform. If `requests-toolbelt` is not installed or the file-like object is not seekable, then it will need to read the entire buffer into memory before writing.

Examples

```
>>> # 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()`

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(source_db, dest_db, ..., ...)</code>	Transfer a table from one location to another.
<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 transfered. 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='schma.tbl', dest_table='schma.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
```

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.

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 to one-hot-encode categorical features (the non-numerical columns) to binary indicator columns before sending the features to the `Pipeline`.

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
<code>sparse_logistic</code>	classification	<code>LogisticRegression</code>	<code>C=499999950, tol=1e-08</code>
<code>gradient_boosting_classifier</code>	classification	<code>GradientBoostingClassifier</code>	<code>n_estimators=500, max_depth=2</code>
<code>random_forest_classifier</code>	classification	<code>RandomForestClassifier</code>	<code>n_estimators=500</code>
<code>extra_trees_classifier</code>	classification	<code>ExtraTreesClassifier</code>	<code>n_estimators=500</code>
<code>sparse_linear_regressor</code>	regression	<code>LinearRegression</code>	
<code>sparse_ridge_regressor</code>	regression	<code>Ridge</code>	
<code>gradient_boosting_regressor</code>	regression	<code>GradientBoostingRegressor</code>	<code>n_estimators=500, max_depth=2</code>
<code>random_forest_regressor</code>	regression	<code>RandomForestRegressor</code>	<code>n_estimators=500</code>
<code>extra_trees_regressor</code>	regression	<code>ExtraTreesRegressor</code>	<code>n_estimators=500</code>

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 be built from libraries which CivisML recognizes. You can use code from

- `scikit-learn` v0.18.1
- `glmnet` v2.0.0
- `xgboost` v0.6a2
- `muffnn` v1.1.1

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 Dependencies

Installing packages from PyPI is straightforward. You can specify a `dependencies` argument to `~civis.ml.ModelPipeline` which will install the dependencies in your runtime environment. VCS support is also enabled (see [docs](https://pip.pypa.io/en/stable/reference/pip_install/#vcs-support)). 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 `~civis.ml.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.

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.

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.

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
```

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

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 custom models or download trained models, you'll need scikit-learn installed.

The “`sparse_logistic`”, “`sparse_linear_regressor`”, and “`sparse_ridge_regressor`” models all use the public Civis Analytics `glmnet` library. Install it if you wish to download a model created from one of these pre-defined models.

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)
```

Interface for scikit-learn modeling in the Civis Platform

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

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.

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, optional

Cross validation parameter grid for learner parameters, e.g. { 'n_estimators' : [100, 200, 500], 'learning_rate' : [0.01, 0.1], 'max_depth' : [2, 3] }.

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).

git_token_name [str, optional] Name of remote git API token stored in 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.

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',
...                      n_jobs=50)
>>> 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

<code>train()</code>	Train the model on data in Civis Platform; outputs <i>ModelFuture</i>
<code>predict()</code>	Make predictions on new data; outputs <i>ModelFuture</i>
<code>from_existing()</code>	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)

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.

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.

polling_interval : float, optional

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

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)
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.

`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.

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.

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

metadata	(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 <code>job_id</code> if this is a training job.
train_run_id	(int) As <code>train_job_id</code> 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.

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.

Joblib backend

`joblib` is a tool which facilitates parallel processing in Python. 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.

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()`. Use the `docker_image_name`, `docker_image_tag`, `repo_http_uri`, and `repo_ref` parameters to define the environment in which your code will be run. Make sure that this environment includes all of the code which you’re parallelizing.

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):

```
>>> 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]
```

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:

```
>>> 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]
```

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.

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, **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.

****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, **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.

****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.

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', re-
sources='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

credentials	An instance of the <i>Credentials</i> endpoint
databases	An instance of the <i>Databases</i> endpoint
files	An instance of the <i>Files</i> endpoint
imports	An instance of the <i>Imports</i> endpoint
jobs	An instance of the <i>Jobs</i> endpoint
models	An instance of the <i>Models</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
reports	An instance of the <i>Reports</i> endpoint
scripts	An instance of the <i>Scripts</i> endpoint
tables	An instance of the <i>Tables</i> endpoint
users	An instance of the <i>Users</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.

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

<i>json_data</i>	(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.
<i>headers</i>	(dict) This is the header for the API call without changing the key names.
<i>calls_remaining</i>	(int) Number of API calls remaining before rate limit is reached.
<i>rate_limit</i>	(int) Total number of calls per API rate limit period.

class `civis.response.PaginatedResponse` (*path*, *initial_params*, *endpoint*)

A response object that supports iteration.

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)

```

API Resources

Credentials

class `Credentials` (*session*, *return_type*='civis')

Methods

<code>get(id)</code>	Get a credential
<code>list(**kwargs)</code>	List credentials
<code>post(username, password, type, **kwargs)</code>	Create or update a credential
<code>post_authenticate(password, username, url, ...)</code>	Authenticate against a remote host
<code>post_temporary(id, **kwargs)</code>	Generate a temporary credential for accessing S3
<code>put(id, username, password, type, **kwargs)</code>	Update an existing credential

get (*id*)

Get a credential

Parameters *id* : integer

The ID of the credential.

Returns *username* : string

The username for the credential.

description : string

A long description of the credential.

remote_host_name : string

The name of the remote host associated with this credential.

id : integer

The ID of the credential.

updated_at : string/time

The last modification time for this credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

type : string

The credential's type.

owner : string

The name of the user who this credential belongs to.

created_at : string/time

The creation time for this credential.

name : string

The name identifying the credential

list (***kwargs*)

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 **username** : string

The username for the credential.

description : string

A long description of the credential.

remote_host_name : string

The name of the remote host associated with this credential.

id : integer

The ID of the credential.

updated_at : string/time

The last modification time for this credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

type : string

The credential's type.

owner : string

The name of the user who this credential belongs to.

created_at : string/time

The creation time for this credential.

name : string

The name identifying the credential

post (*username, password, type, **kwargs*)

Create or update a credential

Parameters **username** : string

The username for the credential.

password : string

The password for the credential.

type : string

remote_host : dict, optional:

```
- 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
- url : string
    The URL to your host.
- name : string
    The human readable name for the remote host.
```

description : string, optional

A long description of the credential.

name : string, optional

The name identifying the credential.

remote_host_id : integer, optional

The ID of the remote host associated with the credential.

Returns **username** : string

The username for the credential.

description : string

A long description of the credential.

remote_host_name : string

The name of the remote host associated with this credential.

id : integer

The ID of the credential.

updated_at : string/time

The last modification time for this credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

type : string

The credential's type.

owner : string

The name of the user who this credential belongs to.

created_at : string/time

The creation time for this credential.

name : string

The name identifying the credential

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

Authenticate against a remote host

Parameters **password** : string

The password for the credential.

username : string

The username for the credential.

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

Returns **username** : string

The username for the credential.

description : string

A long description of the credential.

remote_host_name : string

The name of the remote host associated with this credential.

id : integer

The ID of the credential.

updated_at : string/time

The last modification time for this credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

type : string

The credential's type.

owner : string

The name of the user who this credential belongs to.

created_at : string/time

The creation time for this credential.

name : string

The name identifying the credential

post_temporary (*id*, ***kwargs*)

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 **session_token** : string

The session token identifier.

secret_access_key : string

The secret part of the credential.

access_key : string

The identifier of the credential.

put (*id*, *username*, *password*, *type*, ***kwargs*)

Update an existing credential

Parameters **id** : integer

The ID of the credential.

username : string

The username for the credential.

password : string

The password for the credential.

type : string

remote_host : dict, optional:

```
- 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
- url : string
  The URL to your host.
- name : string
  The human readable name for the remote host.
```

description : string, optional

A long description of the credential.

name : string, optional

The name identifying the credential.

remote_host_id : integer, optional

The ID of the remote host associated with the credential.

Returns username : string

The username for the credential.

description : string

A long description of the credential.

remote_host_name : string

The name of the remote host associated with this credential.

id : integer

The ID of the credential.

updated_at : string/time

The last modification time for this credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

type : string

The credential's type.

owner : string

The name of the user who this credential belongs to.

created_at : string/time

The creation time for this credential.

name : string

The name identifying the credential

Databases

class Databases (*session, 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 *is_active* : boolean

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

security_group_id : string

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

updated_at : string/time

The time this rule was last updated.

remote_host_id : integer

The ID of the database this rule is applied to.

id : integer

The ID of this whitelisted IP address.

subnet_mask : string

The subnet mask that is allowed by this rule.

created_at : string/time

The time this rule was created.

authorized_by : string

The user who authorized this rule.

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 security_group_id : string

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

updated_at : string/time

The time this rule was last updated.

remote_host_id : integer

The ID of the database 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.

id : integer

The ID of this whitelisted IP address.

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 is_active : boolean

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

security_group_id : string

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

updated_at : string/time

The time this rule was last updated.

remote_host_id : integer

The ID of the database this rule is applied to.

id : integer

The ID of this whitelisted IP address.

subnet_mask : string

The subnet mask that is allowed by this rule.

created_at : string/time

The time this rule was created.

authorized_by : string

The user who authorized this rule.

Files

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

Methods

<i>delete_projects</i> (id, project_id)	Remove a Data::S3File 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>get</i> (id)	Get details about a file
<i>list_projects</i> (id)	List the projects a Data::S3File belongs to
<i>list_shares</i> (id)	List users and groups permissioned on this object
<i>post</i> (name, **kwargs)	Initiate an upload of a file into the platform
<i>put_projects</i> (id, project_id)	Add a Data::S3File to a project
<i>put_shares_groups</i> (id, permission_level, ...)	Set the permissions groups has on this object
<i>put_shares_users</i> (id, user_ids, permission_level)	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*)

Get details about a file

Parameters **id** : integer

The ID of the file object.

Returns **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.

file_url : string

The URL that may be used to download the file.

name : string

The file name.

download_url : string

A JSON string containing information about the URL of the file.

created_at : string/date-time

The date and time the file was created.

id : integer

The ID of the file object.

list_projects (*id*)

List the projects a Data::S3File belongs to

Parameters **id** : integer

The ID of the resource.

Returns `archived` : string

The archival status of the requested object(s).

author : dict:

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

description : string

A description of the project

users : list:

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

name : string

The name of this project.

updated_at : string/time

auto_share : boolean

created_at : string/time

id : integer

The ID for this project.

list_shares (*id*)

List users and groups permissioned on this object

Parameters `id` : integer

The ID of the object.

Returns `total_group_shares` : integer

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

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
```

readers : 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.

post (*name*, ***kwargs*)

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 **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.

name : string

The file name.

created_at : string/date-time

The date and time the file was created.

id : integer

The ID of the file object.

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*, *permission_level*, *group_ids*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

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

group_ids : list

An array of one or more group IDs

Returns **total_group_shares** : integer

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

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
```

readers : 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.

put_shares_users (*id*, *user_ids*, *permission_level*)

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”

Returns **total_group_shares** : integer

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

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

```

readers : 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.

Imports

class Imports (*session*, *return_type='civis'*)

Methods

<i>delete_files_runs</i>(id, run_id)	Cancel a run
<i>delete_projects</i>(id, project_id)	Remove a JobTypes::Import 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_syncs</i>(id, sync_id)	Delete a sync
<i>get</i>(id)	Get details about an import
<i>get_batches</i>(id)	Get details about a batch import
<i>get_files_runs</i>(id, run_id)	Check status of a run
<i>list</i>(**kwargs)	List imports
<i>list_batches</i>(**kwargs)	List batch imports
<i>list_files_runs</i>(id, **kwargs)	List runs for the given import
<i>list_projects</i>(id)	List the projects a JobTypes::Import belongs to
<i>list_runs</i>(id)	Get the run history of this import
<i>list_shares</i>(id)	List users and groups permissioned on this object
<i>post</i>(name, is_outbound, sync_type, **kwargs)	Create a new import configuration
<i>post_batches</i>(table, file_ids, schema, ...)	Upload multiple files to Redshift
<i>post_cancel</i>(id)	Cancel a run
<i>post_files</i>(schema, remote_host_id, ...)	Initate an import of a tabular file into the platform
<i>post_files_runs</i>(id)	Start a run
<i>post_runs</i>(id)	Run an import
<i>post_syncs</i>(id, source, destination, **kwargs)	Create a sync
<i>put</i>(id, name, is_outbound, sync_type, **kwargs)	Update an import
<i>put_archive</i>(id, status)	Update the archive status of this object
<i>put_projects</i>(id, project_id)	Add a JobTypes::Import to a project
<i>put_shares_groups</i>(id, permission_level, ...)	Set the permissions groups has on this object
<i>put_shares_users</i>(id, user_ids, permission_level)	Set the permissions users have on this object
<i>put_syncs</i>(id, sync_id, source, destination, ...)	Update a 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

delete_syncs (*id, sync_id*)

Delete a sync

Parameters **id** : integer

The ID of the import to fetch.

sync_id : integer

The ID of the sync to fetch.

Returns None

Response code 204: success

get (*id*)

Get details about an import

Parameters **id** : integer

The ID for the import.

Returns **schedule** : dict:

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

syncs : list:

```
List of syncs.
- advanced_options : dict::
    - sortkey2 : string
    - row_chunk_size : integer
    - partition_column_name : string
    - partition_table_name : string
    - export_action : string
    - existing_table_rows : string
    - verify_table_row_counts : boolean
    - last_modified_column : string
    - partition_schema_name : string
    - truncate_long_lines : boolean
    - partition_table_partition_column_max_name : string
    - column_delimiter : string
    - max_errors : integer
    - wipe_destination_table : boolean
    - identity_column : string
    - soql_query : string
    - partition_table_partition_column_min_name : string
    - contact_lists : string
    - sql_query : string
    - first_row_is_header : boolean
    - distkey : string
    - invalid_char_replacement : string
    - sortkey1 : string
    - mysql_catalog_matches_schema : boolean
- id : integer
- destination : dict::
    - path : string
      The schema.tablename to sync to.
- source : dict::
    - path : string
      The path of the dataset to sync from; for a database source,
      schema.tablename.
    - id : integer
      The ID of the table or file, if available.
```

name : string

The name of the import.

updated_at : string/date-time

state : string

is_outbound : boolean

sync_type : string

The type of sync to perform; one of Dbsync, AutoImport, SilverpopDataImport, SilverpopContactImport, GdocImport, GdocExport, and Salesforce.

created_at : string/date-time

hidden : boolean

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

archived : string

The archival status of the requested object(s).

source : dict:

```
- 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.
- credential_id : integer
- name : string
- remote_host_id : integer
```

destination : dict:

```
- 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.
- credential_id : integer
- name : string
- remote_host_id : integer
```

time_zone : string

The time zone of this import.

parent_id : integer

Parent id to trigger this import from

user : dict:

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

running_as : dict:

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

next_run_at : string/time

The time of the next scheduled run.

last_run : dict:

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

notifications : dict:

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

id : integer

The ID for the import.

get_batches (*id*)

Get details about a batch import

Parameters **id** : integer

The ID for the import.

Returns **started_at** : string/time

The time the last run started at.

table : string

The destination table name, without the schema prefix. This table must already exist in Redshift.

id : integer

The ID for the import.

error : string

The error returned by the run, if any.

remote_host_id : integer

The ID of the destination database host.

hidden : boolean

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

state : string

The state of the run; one of “queued”, “running”, “succeeded”, “failed”, or “cancelled”.

finished_at : string/time

The time the last run completed.

schema : string

The destination schema name. This schema must already exist in Redshift.

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 **started_at** : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

import_id : integer

The ID of the import.

finished_at : string/time

The time the last run completed.

list (***kwargs*)

List imports

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'.

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 **schedule** : dict:

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

```

    Hours of the day it is scheduled on
-   scheduled_minutes : list
    Minutes of the day it is scheduled on

```

name : string

The name of the import.

updated_at : string/date-time

state : string

is_outbound : boolean

sync_type : string

The type of sync to perform; one of DbSync, AutoImport, SilverpopDataImport, SilverpopContactImport, GdocImport, GdocExport, and Salesforce.

created_at : string/date-time

archived : string

The archival status of the requested object(s).

source : dict:

```

-   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.
-   credential_id : integer
-   name : string
-   remote_host_id : integer

```

destination : dict:

```

-   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.
-   credential_id : integer
-   name : string
-   remote_host_id : integer

```

time_zone : string

The time zone of this import.

user : dict:

```

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

```

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

last_run : dict:

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

id : integer

The ID for the import.

list_batches (***kwargs*)

List batch imports

Parameters **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 **started_at** : string/time

The time the last run started at.

table : string

The destination table name, without the schema prefix. This table must already exist in Redshift.

id : integer

The ID for the import.

error : string

The error returned by the run, if any.

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”.

finished_at : string/time

The time the last run completed.

schema : string

The destination schema name. This schema must already exist in Redshift.

list_files_runs (*id*, ***kwargs*)

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 **started_at** : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

import_id : integer

The ID of the import.

finished_at : string/time

The time the last run completed.

list_projects (*id*)

List the projects a JobTypes::Import belongs to

Parameters **id** : integer

The ID of the resource.

Returns **archived** : string

The archival status of the requested object(s).

author : dict:

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

description : string

A description of the project

users : list:

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

name : string

The name of this project.

updated_at : string/time

auto_share : boolean

created_at : string/time

id : integer

The ID for this project.

list_runs (*id*)

Get the run history of this import

Parameters **id** : integer

Returns `started_at` : string/time

The time that the run started.

id : integer

error : string

The error message for this run, if present.

created_at : string/time

The time that the run was queued.

state : string

finished_at : string/time

The time that the run completed.

list_shares (*id*)

List users and groups permissioned on this object

Parameters `id` : integer

The ID of the object.

Returns `total_group_shares` : integer

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

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
```

readers : 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.

post (*name*, *is_outbound*, *sync_type*, ***kwargs*)

Create a new import configuration

Parameters **name** : string

The name of the import.

is_outbound : boolean

sync_type : string

The type of sync to perform; one of Dbsync, AutoImport, SilverpopDataImport, SilverpopContactImport, GdocImport, GdocExport, and Salesforce.

schedule : dict, optional:

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

hidden : boolean, 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

destination : dict, optional:

```
- 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.
- credential_id : integer
- remote_host_id : integer
```

time_zone : string, optional

The time zone of this import.

parent_id : integer, optional

Parent id to trigger this import from

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on
- urls : list
```

```

    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_subject : string
    Custom subject line for success e-mail.

```

source : dict, optional:

```

- 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.
- credential_id : integer
- remote_host_id : integer

```

Returns schedule : dict:

```

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

```

syncs : list:

```

List of syncs.
- advanced_options : dict::
    - sortkey2 : string
    - row_chunk_size : integer
    - partition_column_name : string
    - partition_table_name : string
    - export_action : string
    - existing_table_rows : string
    - verify_table_row_counts : boolean
    - last_modified_column : string
    - partition_schema_name : string
    - truncate_long_lines : boolean
    - partition_table_partition_column_max_name : string
    - column_delimiter : string
    - max_errors : integer
    - wipe_destination_table : boolean
    - identity_column : string
    - soql_query : string

```

```
- partition_table_partition_column_min_name : string
- contact_lists : string
- sql_query : string
- first_row_is_header : boolean
- distkey : string
- invalid_char_replacement : string
- sortkey1 : string
- mysql_catalog_matches_schema : boolean
- id : integer
- destination : dict::
    - path : string
      The schema.tablename to sync to.
- source : dict::
    - path : string
      The path of the dataset to sync from; for a database source,
      schema.tablename.
    - id : integer
      The ID of the table or file, if available.
```

name : string

The name of the import.

updated_at : string/date-time

state : string

is_outbound : boolean

sync_type : string

The type of sync to perform; one of Dbsync, AutoImport, SilverpopDataImport, SilverpopContactImport, GdocImport, GdocExport, and Salesforce.

created_at : string/date-time

hidden : boolean

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

archived : string

The archival status of the requested object(s).

source : dict:

```
- 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.
- credential_id : integer
- name : string
- remote_host_id : integer
```

destination : dict:

```
- 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.
- credential_id : integer
- name : string
- remote_host_id : integer

```

time_zone : string

The time zone of this import.

parent_id : integer

Parent id to trigger this import from

user : dict:

```

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

```

running_as : dict:

```

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

```

next_run_at : string/time

The time of the next scheduled run.

last_run : dict:

```

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

```

notifications : dict:

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

id : integer

The ID for the import.

post_batches (*table, file_ids, schema, remote_host_id, credential_id, **kwargs*)

Upload multiple files to Redshift

Parameters **table** : string

The destination table name, without the schema prefix. This table must already exist in Redshift.

file_ids : list

The file IDs for the import.

schema : string

The destination schema name. This schema 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.

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.

hidden : boolean, 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

column_delimiter : string, optional

The column delimiter for the file. Valid arguments are “comma”, “tab”, and “pipe”. If unspecified, defaults to “comma”.

compression : string, optional

The type of compression. Valid arguments are “gzip”, “zip”, and “none”. If unspecified, defaults to “gzip”.

Returns **started_at** : string/time

The time the last run started at.

table : string

The destination table name, without the schema prefix. This table must already exist in Redshift.

id : integer

The ID for the import.

error : string

The error returned by the run, if any.

remote_host_id : integer

The ID of the destination database host.

hidden : boolean

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

state : string

The state of the run; one of “queued”, “running”, “succeeded”, “failed”, or “cancelled”.

finished_at : string/time

The time the last run completed.

schema : string

The destination schema name. This schema must already exist in Redshift.

post_cancel (*id*)

Cancel a run

Parameters **id** : integer

The ID of the job.

Returns **state** : string

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

id : integer

The ID of the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

post_files (*schema, remote_host_id, credential_id, name, **kwargs*)

Initiate an import of a tabular file into the platform

Parameters **schema** : string

The schema 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.

name : string

The name of the destination 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”.

hidden : boolean, 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

multipart : boolean, optional

If true, the upload URI will require a *multipart/form-data* POST request. Defaults to false.

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.

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.

sortkey2 : string, optional

The second column in a compound sortkey for the table.

max_errors : integer, optional

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

sortkey1 : string, optional

The column to use as the sort key for the table.

Returns **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.

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.

post_files_runs (*id*)

Start a run

Parameters *id* : integer

The ID of the import.

Returns *started_at* : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

import_id : integer

The ID of the import.

finished_at : string/time

The time the last run completed.

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*, ***kwargs*)

Create a sync

Parameters *id* : integer**source** : dict:

```
- path : string
    The path of the dataset to sync from; for a database source,
    schema.tablename.
```

destination : dict:

```
- path : string
    The schema.tablename to sync to.
```

advanced_options : dict, optional:

```
- sortkey2 : string
- row_chunk_size : integer
- partition_column_name : string
- partition_table_name : string
- export_action : string
- existing_table_rows : string
- verify_table_row_counts : boolean
- last_modified_column : string
- partition_schema_name : string
- truncate_long_lines : boolean
- partition_table_partition_column_max_name : string
- column_delimiter : string
- max_errors : integer
- wipe_destination_table : boolean
- identity_column : string
- soql_query : string
- partition_table_partition_column_min_name : string
- contact_lists : string
- sql_query : string
- first_row_is_header : boolean
- distkey : string
- invalid_char_replacement : string
- sortkey1 : string
- mysql_catalog_matches_schema : boolean
```

Returns `advanced_options` : dict:

```
- sortkey2 : string
- row_chunk_size : integer
- partition_column_name : string
- partition_table_name : string
- export_action : string
- existing_table_rows : string
- verify_table_row_counts : boolean
- last_modified_column : string
- partition_schema_name : string
- truncate_long_lines : boolean
- partition_table_partition_column_max_name : string
- column_delimiter : string
- max_errors : integer
- wipe_destination_table : boolean
- identity_column : string
- soql_query : string
- partition_table_partition_column_min_name : string
- contact_lists : string
- sql_query : string
- first_row_is_header : boolean
- distkey : string
- invalid_char_replacement : string
- sortkey1 : string
- mysql_catalog_matches_schema : boolean
```

id : integer

destination : dict:

```
- path : string
    The schema.tablename to sync to.
```

source : dict:

```
- path : string
    The path of the dataset to sync from; for a database source,
    schema.tablename.
- id : integer
    The ID of the table or file, if available.
```

put (*id, name, is_outbound, sync_type, **kwargs*)
Update an import

Parameters **id** : integer

The ID for the import.

name : string

The name of the import.

is_outbound : boolean

sync_type : string

The type of sync to perform; one of Dbsync, AutoImport, SilverpopDataImport, SilverpopContactImport, GdocImport, GdocExport, and Salesforce.

schedule : dict, optional:

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

destination : dict, optional:

```
- 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.
- credential_id : integer
- remote_host_id : integer
```

time_zone : string, optional

The time zone of this import.

parent_id : integer, optional

Parent id to trigger this import from

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

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

source : dict, optional:

```
- 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.
- credential_id : integer
- remote_host_id : integer
```

Returns schedule : dict:

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

syncs : list:

```
List of syncs.
- advanced_options : dict::
    - sortkey2 : string
    - row_chunk_size : integer
    - partition_column_name : string
    - partition_table_name : string
    - export_action : string
    - existing_table_rows : string
    - verify_table_row_counts : boolean
    - last_modified_column : string
    - partition_schema_name : string
    - truncate_long_lines : boolean
```

```

- partition_table_partition_column_max_name : string
- column_delimiter : string
- max_errors : integer
- wipe_destination_table : boolean
- identity_column : string
- soql_query : string
- partition_table_partition_column_min_name : string
- contact_lists : string
- sql_query : string
- first_row_is_header : boolean
- distkey : string
- invalid_char_replacement : string
- sortkey1 : string
- mysql_catalog_matches_schema : boolean
- id : integer
- destination : dict::
    - path : string
      The schema.tablename to sync to.
- source : dict::
    - path : string
      The path of the dataset to sync from; for a database source,
      schema.tablename.
    - id : integer
      The ID of the table or file, if available.

```

name : string

The name of the import.

updated_at : string/date-time

state : string

is_outbound : boolean

sync_type : string

The type of sync to perform; one of Dbsync, AutoImport, SilverpopDataImport, SilverpopContactImport, GdocImport, GdocExport, and Salesforce.

created_at : string/date-time

hidden : boolean

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

archived : string

The archival status of the requested object(s).

source : dict:

```

- 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.
- credential_id : integer
- name : string
- remote_host_id : integer

```

destination : dict:

```
- 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.
- credential_id : integer
- name : string
- remote_host_id : integer
```

time_zone : string

The time zone of this import.

parent_id : integer

Parent id to trigger this import from

user : dict:

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

running_as : dict:

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

next_run_at : string/time

The time of the next scheduled run.

last_run : dict:

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

```
- finished_at : string/time
    The time that the run completed.
```

notifications : dict:

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

id : integer

The ID for the import.

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 **schedule** : dict:

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

syncs : list:

```
List of syncs.
- advanced_options : dict::
    - sortkey2 : string
    - row_chunk_size : integer
    - partition_column_name : string
    - partition_table_name : string
```

```
- export_action : string
- existing_table_rows : string
- verify_table_row_counts : boolean
- last_modified_column : string
- partition_schema_name : string
- truncate_long_lines : boolean
- partition_table_partition_column_max_name : string
- column_delimiter : string
- max_errors : integer
- wipe_destination_table : boolean
- identity_column : string
- soql_query : string
- partition_table_partition_column_min_name : string
- contact_lists : string
- sql_query : string
- first_row_is_header : boolean
- distkey : string
- invalid_char_replacement : string
- sortkey1 : string
- mysql_catalog_matches_schema : boolean
- id : integer
- destination : dict::
  - path : string
    The schema.tablename to sync to.
- source : dict::
  - path : string
    The path of the dataset to sync from; for a database source,
    schema.tablename.
  - id : integer
    The ID of the table or file, if available.
```

name : string

The name of the import.

updated_at : string/date-time

state : string

is_outbound : boolean

sync_type : string

The type of sync to perform; one of Dbsync, AutoImport, SilverpopDataImport, SilverpopContactImport, GdocImport, GdocExport, and Salesforce.

created_at : string/date-time

hidden : boolean

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

archived : string

The archival status of the requested object(s).

source : dict:

```
- 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.
- credential_id : integer
- name : string
- remote_host_id : integer

```

destination : dict:

```

- 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.
- credential_id : integer
- name : string
- remote_host_id : integer

```

time_zone : string

The time zone of this import.

parent_id : integer

Parent id to trigger this import from

user : dict:

```

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

```

running_as : dict:

```

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

```

next_run_at : string/time

The time of the next scheduled run.

last_run : dict:

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

notifications : dict:

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

id : integer

The ID for the import.

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*, *permission_level*, *group_ids*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

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

group_ids : list

An array of one or more group IDs

Returns `total_group_shares` : integer

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

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
```

readers : 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.

put_shares_users (*id*, *user_ids*, *permission_level*)

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”

Returns `total_group_shares` : integer

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

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
```

readers : 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.

put_syncs (*id*, *sync_id*, *source*, *destination*, ***kwargs*)

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.
```

destination : dict:

```
- path : string  
  The schema.tablename to sync to.
```

advanced_options : dict, optional:

```
- sortkey2 : string  
- row_chunk_size : integer  
- partition_column_name : string  
- partition_table_name : string  
- export_action : string  
- existing_table_rows : string
```

```
- verify_table_row_counts : boolean
- last_modified_column : string
- partition_schema_name : string
- truncate_long_lines : boolean
- partition_table_partition_column_max_name : string
- column_delimiter : string
- max_errors : integer
- wipe_destination_table : boolean
- identity_column : string
- soql_query : string
- partition_table_partition_column_min_name : string
- contact_lists : string
- sql_query : string
- first_row_is_header : boolean
- distkey : string
- invalid_char_replacement : string
- sortkey1 : string
- mysql_catalog_matches_schema : boolean
```

Returns `advanced_options` : dict:

```
- sortkey2 : string
- row_chunk_size : integer
- partition_column_name : string
- partition_table_name : string
- export_action : string
- existing_table_rows : string
- verify_table_row_counts : boolean
- last_modified_column : string
- partition_schema_name : string
- truncate_long_lines : boolean
- partition_table_partition_column_max_name : string
- column_delimiter : string
- max_errors : integer
- wipe_destination_table : boolean
- identity_column : string
- soql_query : string
- partition_table_partition_column_min_name : string
- contact_lists : string
- sql_query : string
- first_row_is_header : boolean
- distkey : string
- invalid_char_replacement : string
- sortkey1 : string
- mysql_catalog_matches_schema : boolean
```

id : integer

destination : dict:

```
- path : string
    The schema.tablename to sync to.
```

source : dict:

```
- path : string
    The path of the dataset to sync from; for a database source,
    schema.tablename.
```

```
- id : integer
    The ID of the table or file, if available.
```

Jobs

class Jobs (*session*, *return_type*='civis')

Methods

<i>delete_projects</i> (id, project_id)	Remove a Job 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>get</i> (id)	Show basic job info
<i>get_runs</i> (id, run_id)	Check status of a job
<i>list</i> (**kwargs)	List jobs
<i>list_children</i> (id)	Show nested tree of children that this job triggers
<i>list_parents</i> (id)	Show chain of parents as a list that this job triggers from
<i>list_projects</i> (id)	List the projects a Job belongs to
<i>list_shares</i> (id)	List users and groups permissioned on this object
<i>post_runs</i> (id)	Run a job
<i>post_trigger_email</i> (id)	Generate and retrieve trigger email address
<i>put_projects</i> (id, project_id)	Add a Job to a project
<i>put_shares_groups</i> (id, permission_level, ...)	Set the permissions groups has on this object
<i>put_shares_users</i> (id, user_ids, permission_level)	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 `archived` : string

The archival status of the requested object(s).

runs : list:

```

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

```

hidden : boolean

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

id : integer**updated_at** : string/date-time**type** : string**state** : string

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

last_run : dict:

```

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

```

created_at : string/date-time

name : string

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 **started_at** : string/time

The time that the run started.

id : integer

error : string

The error message for this run, if present.

created_at : string/time

The time that the run was queued.

state : string

finished_at : string/time

The time that the run completed.

list (***kwargs*)

List jobs

Parameters **limit** : integer, optional

The maximum number of jobs to return.

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.

archived : string, optional

The archival status of the requested object(s).

Returns **archived** : string

The archival status of the requested object(s).

id : integer**updated_at** : string/date-time**type** : string**state** : string

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

last_run : dict:

```

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

```

created_at : string/date-time**name** : string**list_children** (*id*)

Show nested tree of children that this job triggers

Parameters **id** : integer

The ID for this job.

Returns **runs** : list:

```

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

```

id : integer**updated_at** : string/date-time**type** : string**state** : string**children** : list**last_run** : dict:

```

- started_at : string/time
    The time that the run started.
- id : integer
- error : string
    The error message for this run, if present.

```

```
- created_at : string/time
    The time that the run was queued.
- state : string
- finished_at : string/time
    The time that the run completed.
```

created_at : string/date-time

name : string

list_parents (*id*)

Show chain of parents as a list that this job triggers from

Parameters **id** : integer

The ID for this job.

Returns **archived** : string

The archival status of the requested object(s).

runs : list:

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

hidden : boolean

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

id : integer

updated_at : string/date-time

type : string

state : string

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

last_run : dict:

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

created_at : string/date-time

name : string

list_projects (*id*)

List the projects a Job belongs to

Parameters **id** : integer

The ID of the resource.

Returns **archived** : string

The archival status of the requested object(s).

author : dict:

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

description : string

A description of the project

users : list:

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

name : string

The name of this project.

updated_at : string/time

auto_share : boolean

created_at : string/time

id : integer

The ID for this project.

list_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns `total_group_shares` : integer

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

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
```

readers : 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.

post_runs (*id*)

Run a job

Parameters `id` : integer

The ID for this job.

Returns `started_at` : string/time

The time that the run started.

id : integer

error : string

The error message for this run, if present.

created_at : string/time

The time that the run was queued.

state : string

finished_at : string/time

The time that the run completed.

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*, *permission_level*, *group_ids*)

Set the permissions groups has on this object

Parameters *id* : integer

ID of the resource to be shared

permission_level : string

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

group_ids : list

An array of one or more group IDs

Returns *total_group_shares* : integer

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

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
```

readers : 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.

put_shares_users (*id*, *user_ids*, *permission_level*)

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”

Returns **total_group_shares** : integer

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

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
```

readers : 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.

Models

class Models (*session*, *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(**kwargs)</code>	List models
<code>list_builds(id, **kwargs)</code>	List builds for the given model
<code>list_projects(id)</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
<code>list_types()</code>	List all available model types
<code>patch(id, **kwargs)</code>	Update model configuration
<code>post(**kwargs)</code>	Create new configuration for a model
<code>post_builds(id)</code>	Start a build
<code>put_archive(id, status)</code>	Update the archive status of this object
<code>put_predictions(id, primary_key, table_name, ...)</code>	Add a table on which to apply the predictive model
<code>put_projects(id, project_id)</code>	Add a models to a project
<code>put_schedules(id, schedule)</code>	Schedule the model build
<code>put_shares_groups(id, permission_level, ...)</code>	Set the permissions groups has on this object
<code>put_shares_users(id, user_ids, permission_level)</code>	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 **schedule** : dict:

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

database_id : integer

The ID of the database holding the training set table used to build the model.

description : string

A description of the model.

hidden : boolean

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

id : integer

The ID of the model.

dependent_variable : string

The dependent variable of the training dataset.

credential_id : integer

The ID of the credential used to read the target table. Defaults to the user's default credential.

last_output_location : string

The output JSON for the last build.

created_at : string/date-time

The time the model was created.

interaction_terms : boolean

Whether to search for interaction terms.

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]}.

archived : string

The archival status of the requested object(s).

active_build_id : integer

The ID of the current active build, the build used to score predictions.

box_cox_transformation : boolean

Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.

parent_id : integer

The ID of the parent job that will trigger this model.

number_of_folds : integer

Number of folds for cross validation. Default value is 5.

predictions : list:

The tables upon which the model will be applied.

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

```
    The qualified name of the table to be created which will_
↪ contain the
    model's predictions.
- id : integer
    The ID of the model to which to apply the prediction.
- primary_key : list
    The primary key or composite keys of the table being predicted.
- state : string
    The status of the prediction. One of: "succeeded", "failed",
↪ "queued",
    or "running", or "idle", if no build has been attempted.
- limiting_sql : string
    A SQL WHERE clause used to scope the rows to be predicted.
- table_name : string
    The qualified name of the table on which to apply the_
↪ predictive model.
```

model_type_id : integer

The ID of the model's type.

last_run : dict:

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

limiting_sql : string

A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., "id > 105").

updated_at : string/date-time

The time the model was updated.

time_zone : string

The time zone of this model.

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.

user : dict:

```
- username : string
    This user's username.
- initials : string
    This user's initials.
```

```

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

```

dependent_variable_order : list

The order of dependent variables, especially useful for Ordinal Modeling.

model_name : string

The name of the model.

table_name : string

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

builds : list:

A **list** of trained models available **for** making predictions.

```

- roc_auc : number/float
    A key metric for binary, multinomial, and ordinal models. Nil for other
    model types.
- description : string
    A description of the model build.
- id : integer
    The ID of the model build.
- name : string
    The name of the model build.
- r_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
- created_at : string
    The time the model build was created.
- root_mean_squared_error : number/float
    A key metric for continuous models. Nil for other model types.

```

primary_key : string

The unique ID (primary key) of the training dataset.

excluded_columns : list

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

running_as : dict:

```

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

```

notifications : dict:

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

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 **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.

roc_auc : number/float

A key metric for binary, multinomial, and ordinal models. Nil for other model types.

description : string

A description of the model build.

id : integer

The ID of the model build.

error : string

The error, if any, returned by the build.

state : string

The state of the model build. one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

name : string

The name of the model build.

output : string

A string representing the JSON output for the specified build. Only present when smaller than 10KB in size.

created_at : string

The time the model build was created.

transformation_metadata : string

A string representing the full JSON output of the metadata for transformation of column names

r_squared_error : number/float

A key metric for continuous models. Nil for other model types.

root_mean_squared_error : number/float

A key metric for continuous models. Nil for other model types.

list (***kwargs*)

List models

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’.

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, model_name, created_at, name, 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 **schedule** : dict:

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

database_id : integer

The ID of the database holding the training set table used to build the model.

description : string

A description of the model.

id : integer

The ID of the model.

dependent_variable : string

The dependent variable of the training dataset.

credential_id : integer

The ID of the credential used to read the target table. Defaults to the user's default credential.

last_output_location : string

The output JSON for the last build.

created_at : string/date-time

The time the model was created.

interaction_terms : boolean

Whether to search for interaction terms.

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]}.

archived : string

The archival status of the requested object(s).

box_cox_transformation : boolean

Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.

parent_id : integer

The ID of the parent job that will trigger this model.

number_of_folds : integer

Number of folds for cross validation. Default value is 5.

excluded_columns : list

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

model_type_id : integer

The ID of the model's type.

last_run : dict:

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

limiting_sql : string

A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., "id > 105").

predictions : list:

```
The tables upon which the model will be applied.
- output_table : string
    The qualified name of the table to be created which will_
↪ contain the
    model's predictions.
- id : integer
    The ID of the model to which to apply the prediction.
- primary_key : list
    The primary key or composite keys of the table being predicted.
- state : string
    The status of the prediction. One of: "succeeded", "failed",
↪ "queued",
    or "running", or "idle", if no build has been attempted.
- limiting_sql : string
    A SQL WHERE clause used to scope the rows to be predicted.
- table_name : string
    The qualified name of the table on which to apply the_
↪ predictive model.
```

updated_at : string/date-time

The time the model was updated.

time_zone : string

The time zone of this model.

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.

dependent_variable_order : list

The order of dependent variables, especially useful for Ordinal Modeling.

model_name : string

The name of the model.

table_name : string

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

builds : list:

```
A list of trained models available for making predictions.
- roc_auc : number/float
    A key metric for binary, multinomial, and ordinal models. Nil
    for other model types.
- description : string
    A description of the model build.
- id : integer
    The ID of the model build.
- name : string
    The name of the model build.
- r_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
- created_at : string
    The time the model build was created.
- root_mean_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
```

primary_key : string

The unique ID (primary key) of the training dataset.

user : dict:

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

list_builds (*id*, ***kwargs*)

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 **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.

roc_auc : number/float

A key metric for binary, multinomial, and ordinal models. Nil for other model types.

description : string

A description of the model build.

id : integer

The ID of the model build.

error : string

The error, if any, returned by the build.

state : string

The state of the model build. one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

name : string

The name of the model build.

output : string

A string representing the JSON output for the specified build. Only present when smaller than 10KB in size.

created_at : string

The time the model build was created.

transformation_metadata : string

A string representing the full JSON output of the metadata for transformation of column names

r_squared_error : number/float

A key metric for continuous models. Nil for other model types.

root_mean_squared_error : number/float

A key metric for continuous models. Nil for other model types.

list_projects (*id*)

List the projects a models belongs to

Parameters **id** : integer

The ID of the resource.

Returns **archived** : string

The archival status of the requested object(s).

author : dict:

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

description : string

A description of the project

users : list:

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

name : string

The name of this project.

updated_at : string/time

auto_share : boolean

created_at : string/time

id : integer

The ID for this project.

list_schedules (*id*)

Show the model build schedule

Parameters `id` : integer

The ID of the model associated with this schedule.

Returns `schedule` : dict:

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

id : integer

The ID of the model associated with this schedule.

list_shares (*id*)

List users and groups permissioned on this object

Parameters `id` : integer

The ID of the object.

Returns `total_group_shares` : integer

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

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
```

readers : 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.

list_types()

List all available model types

Returns **int_allowed** : boolean

Whether this model type supports searching for interaction terms.

dv_type : string

The type of dependent variable predicted by the model.

id : integer

The ID of the model type.

algorithm : string

The name of the algorithm used to train the model.

patch(*id*, ***kwargs*)

Update model configuration

Parameters **id** : integer

The ID of the model.

schedule : dict, optional:

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

database_id : integer, optional

The ID of the database holding the training set table used to build the model.

description : string, optional

A description of the model.

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]}.

credential_id : integer, optional

The ID of the credential used to read the target table. Defaults to the user's default credential.

parent_id : integer, optional

The ID of the parent job that will trigger this model.

dependent_variable : string, optional

The dependent variable of the training dataset.

time_zone : string, optional

The time zone of this model.

interaction_terms : boolean, optional

Whether to search for interaction terms.

dependent_variable_order : list, optional

The order of dependent variables, especially useful for Ordinal Modeling.

model_name : string, optional

The name of the model.

table_name : string, optional

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

active_build_id : integer, optional

The ID of the current active build, the build used to score predictions.

box_cox_transformation : boolean, optional

Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.

primary_key : string, optional

The unique ID (primary key) of the training dataset.

excluded_columns : list, optional

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

number_of_folds : integer, optional

Number of folds for cross validation. Default value is 5.

notifications : dict, optional:

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

model_type_id : integer, optional

The ID of the model's type.

limiting_sql : string, optional

A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., “id > 105”).

Returns None

Response code 204: success

post (***kwargs*)

Create new configuration for a model

Parameters **schedule** : dict, optional:

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

database_id : integer, optional

The ID of the database holding the training set table used to build the model.

description : string, optional

A description of the model.

hidden : boolean, 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

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]}.

credential_id : integer, optional

The ID of the credential used to read the target table. Defaults to the user’s default credential.

parent_id : integer, optional

The ID of the parent job that will trigger this model.

dependent_variable : string, optional

The dependent variable of the training dataset.

time_zone : string, optional

The time zone of this model.

interaction_terms : boolean, optional

Whether to search for interaction terms.

dependent_variable_order : list, optional

The order of dependent variables, especially useful for Ordinal Modeling.

model_name : string, optional

The name of the model.

table_name : string, optional

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

active_build_id : integer, optional

The ID of the current active build, the build used to score predictions.

box_cox_transformation : boolean, optional

Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.

primary_key : string, optional

The unique ID (primary key) of the training dataset.

excluded_columns : list, optional

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

number_of_folds : integer, optional

Number of folds for cross validation. Default value is 5.

notifications : dict, optional:

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

model_type_id : integer, optional

The ID of the model's type.

limiting_sql : string, optional

A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., "id > 105").

Returns `schedule` : dict:

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

database_id : integer

The ID of the database holding the training set table used to build the model.

description : string

A description of the model.

hidden : boolean

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

id : integer

The ID of the model.

dependent_variable : string

The dependent variable of the training dataset.

credential_id : integer

The ID of the credential used to read the target table. Defaults to the user's default credential.

last_output_location : string

The output JSON for the last build.

created_at : string/date-time

The time the model was created.

interaction_terms : boolean

Whether to search for interaction terms.

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]}.

archived : string

The archival status of the requested object(s).

active_build_id : integer

The ID of the current active build, the build used to score predictions.

box_cox_transformation : boolean

Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.

parent_id : integer

The ID of the parent job that will trigger this model.

number_of_folds : integer

Number of folds for cross validation. Default value is 5.

predictions : list:

The tables upon which the model will be applied.

```
- schedule : dict::
  - scheduled : boolean
    If the object is scheduled
  - scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
  - scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run
    ↪per hour
  - scheduled_hours : list
    Hours of the day it is scheduled on
  - scheduled_minutes : list
    Minutes of the day it is scheduled on
- output_table : string
  The qualified name of the table to be created which will
  ↪contain the
  model's predictions.
- id : integer
  The ID of the model to which to apply the prediction.
- primary_key : list
  The primary key or composite keys of the table being predicted.
- state : string
  The status of the prediction. One of: "succeeded", "failed",
  ↪"queued",
  or "running", or "idle", if no build has been attempted.
- limiting_sql : string
  A SQL WHERE clause used to scope the rows to be predicted.
- table_name : string
  The qualified name of the table on which to apply the
  ↪predictive model.
```

model_type_id : integer

The ID of the model's type.

last_run : dict:

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

limiting_sql : string

A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., “id > 105”).

updated_at : string/date-time

The time the model was updated.

time_zone : string

The time zone of this model.

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.

user : dict:

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

dependent_variable_order : list

The order of dependent variables, especially useful for Ordinal Modeling.

model_name : string

The name of the model.

table_name : string

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

builds : list:

```
A list of trained models available for making predictions.
- roc_auc : number/float
    A key metric for binary, multinomial, and ordinal models. Nil
    ↪ for other
    model types.
- description : string
    A description of the model build.
- id : integer
    The ID of the model build.
- name : string
    The name of the model build.
- r_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
- created_at : string
    The time the model build was created.
```

```
- root_mean_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
```

primary_key : string

The unique ID (primary key) of the training dataset.

excluded_columns : list

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

running_as : dict:

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

notifications : dict:

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

post_builds (*id*)

Start a build

Parameters *id* : integer

The ID of the model.

Returns *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.

roc_auc : number/float

A key metric for binary, multinomial, and ordinal models. Nil for other model types.

description : string

A description of the model build.

id : integer

The ID of the model build.

error : string

The error, if any, returned by the build.

state : string

The state of the model build. one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

name : string

The name of the model build.

output : string

A string representing the JSON output for the specified build. Only present when smaller than 10KB in size.

created_at : string

The time the model build was created.

transformation_metadata : string

A string representing the full JSON output of the metadata for transformation of column names

r_squared_error : number/float

A key metric for continuous models. Nil for other model types.

root_mean_squared_error : number/float

A key metric for continuous models. Nil for other model types.

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 **schedule** : dict:

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

database_id : integer

The ID of the database holding the training set table used to build the model.

description : string

A description of the model.

hidden : boolean

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

id : integer

The ID of the model.

dependent_variable : string

The dependent variable of the training dataset.

credential_id : integer

The ID of the credential used to read the target table. Defaults to the user's default credential.

last_output_location : string

The output JSON for the last build.

created_at : string/date-time

The time the model was created.

interaction_terms : boolean

Whether to search for interaction terms.

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]}.

archived : string

The archival status of the requested object(s).

active_build_id : integer

The ID of the current active build, the build used to score predictions.

box_cox_transformation : boolean

Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.

parent_id : integer

The ID of the parent job that will trigger this model.

number_of_folds : integer

Number of folds for cross validation. Default value is 5.

predictions : list:

```
The tables upon which the model will be applied.
- schedule : dict::
  - scheduled : boolean
    If the object is scheduled
  - scheduled_days : list
```

```
    Day based on numeric value starting at 0 for Sunday
    - scheduled_runs_per_hour : integer
      Alternative to scheduled minutes, number of times to run_
↳per hour
    - scheduled_hours : list
      Hours of the day it is scheduled on
    - scheduled_minutes : list
      Minutes of the day it is scheduled on
  - output_table : string
    The qualified name of the table to be created which will_
↳contain the
    model's predictions.
  - id : integer
    The ID of the model to which to apply the prediction.
  - primary_key : list
    The primary key or composite keys of the table being predicted.
  - state : string
    The status of the prediction. One of: "succeeded", "failed",
↳"queued",
    or "running", or "idle", if no build has been attempted.
  - limiting_sql : string
    A SQL WHERE clause used to scope the rows to be predicted.
  - table_name : string
    The qualified name of the table on which to apply the_
↳predictive model.
```

model_type_id : integer

The ID of the model's type.

last_run : dict:

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

limiting_sql : string

A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., "id > 105").

updated_at : string/date-time

The time the model was updated.

time_zone : string

The time zone of this model.

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.

user : dict:

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

dependent_variable_order : list

The order of dependent variables, especially useful for Ordinal Modeling.

model_name : string

The name of the model.

table_name : string

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

builds : list:

```
A list of trained models available for making predictions.
- roc_auc : number/float
    A key metric for binary, multinomial, and ordinal models. Nil for other
    model types.
- description : string
    A description of the model build.
- id : integer
    The ID of the model build.
- name : string
    The name of the model build.
- r_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
- created_at : string
    The time the model build was created.
- root_mean_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
```

primary_key : string

The unique ID (primary key) of the training dataset.

excluded_columns : list

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

running_as : dict:

```
- username : string
    This user's username.
- initials : string
```

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

notifications : dict:

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

put_predictions (*id*, *primary_key*, *table_name*, ***kwargs*)

Add a table on which to apply the predictive model

Parameters *id* : integer

The ID of the model to which to apply the prediction.

primary_key : list

The primary key or composite keys of the table being predicted.

table_name : string

The qualified name of the table on which to apply the predictive model.

schedule : dict, optional:

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

output_table : string, optional

The qualified name of the table to be created which will contain the model's predictions.

limiting_sql : string, optional

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

Returns **schedule** : dict:

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

output_table : string

The qualified name of the table to be created which will contain the model's predictions.

id : integer

The ID of the model to which to apply the prediction.

primary_key : list

The primary key or composite keys of the table being predicted.

state : string

The status of the prediction. One of: "succeeded", "failed", "queued", or "running," or "idle", if no build has been attempted.

limiting_sql : string

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

table_name : string

The qualified name of the table on which to apply the predictive model.

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_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

Returns `schedule` : dict:

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

id : integer

The ID of the model associated with this schedule.

put_shares_groups (*id*, *permission_level*, *group_ids*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

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

group_ids : list

An array of one or more group IDs

Returns **total_group_shares** : integer

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

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
```

readers : 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.

put_shares_users (*id*, *user_ids*, *permission_level*)
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”

Returns **total_group_shares** : integer

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

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
```

readers : 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.

Predictions

class Predictions (*session*, *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(**kwargs)</code>	List predictions
<code>list_runs(id, **kwargs)</code>	List runs for the given prediction
<code>list_schedules(id)</code>	Show the prediction schedule
<code>patch(id, **kwargs)</code>	Update a prediction
<code>post_runs(id)</code>	Start a run
<code>put_schedules(id, **kwargs)</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 **started_at** : string/date-time

The start time of the last run of this prediction.

schedule : dict:

```

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

```

scored_table_name : string

The name of the source table for this prediction.

id : integer

The ID of the prediction.

error : string

The error, if any, of the last run of this prediction.

model_id : integer

The ID of the model used for this prediction.

state : string

The state of the last run of this prediction.

finished_at : string/date-time

The end time of the last run of this prediction.

output_table_name : string

The name of the output table for this prediction.

scored_table_id : integer

The ID of the source table for this prediction.

primary_key : list

The primary key or composite keys of the table being predicted.

scored_tables : list:

```

An array of created prediction tables.
- score_stats : list::
    An array of metrics on the created predictions.
    - avg_score : number/float
        The average score.
    - min_score : number/float
        The minimum score.
    - max_score : number/float
        The maximum score.
    - histogram : list
        The histogram of the distribution of scores.
    - score_name : string
        The name of the score.
- name : string

```

```
The name of table with created predictions.
- id : integer
  The ID of the table with created predictions.
- created_at : string/date-time
  The time when the table with created predictions was created.
- schema : string
  The schema of table with created predictions.
```

last_run : dict:

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

limiting_sql : string

A SQL WHERE clause used to scope the rows to be 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 **prediction_id** : integer

The ID of the prediction.

score_stats : list:

```
An array of metrics on the created predictions.
- avg_score : number/float
  The average score.
- min_score : number/float
  The minimum score.
- max_score : number/float
  The maximum score.
- histogram : list
  The histogram of the distribution of scores.
- score_name : string
  The name of the score.
```

name : string

The name of table created by this predictions run.

state : string

The state of the prediction run.

exception : string

The exception, if any, returned by the prediction run.

created_at : string/date-time

The time when the table with created predictions was created.

id : integer

The ID of the prediction run.

list (***kwargs*)

List predictions

Parameters **model_id** : integer, optional

If specified, only return predictions associated with this model ID.

Returns **output_table_name** : string

The name of the output table for this prediction.

started_at : string/date-time

The start time of the last run of this prediction.

scored_table_name : string

The name of the source table for this prediction.

id : integer

The ID of the prediction.

error : string

The error, if any, of the last run of this prediction.

model_id : integer

The ID of the model used for this prediction.

state : string

The state 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:

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

scored_table_id : integer

The ID of the source table for this prediction.

list_runs (*id*, ***kwargs*)

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 `prediction_id` : integer

The ID of the prediction.

score_stats : list:

```
An array of metrics on the created predictions.
- avg_score : number/float
    The average score.
- min_score : number/float
    The minimum score.
- max_score : number/float
    The maximum score.
- histogram : list
    The histogram of the distribution of scores.
- score_name : string
    The name of the score.
```

name : string

The name of table created by this predictions run.

state : string

The state of the prediction run.

exception : string

The exception, if any, returned by the prediction run.

created_at : string/date-time

The time when the table with created predictions was created.

id : integer

The ID of the prediction run.

list_schedules (*id*)

Show the prediction schedule

Parameters `id` : integer

ID of the prediction associated with this schedule.

Returns `schedule` : dict:

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

id : integer

ID of the prediction associated with this schedule.

score_on_model_build : boolean

Whether the prediction will run after a rebuild of the associated model.

patch (*id*, ***kwargs*)

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.

primary_key : list, optional

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.

Returns `started_at` : string/date-time

The start time of the last run of this prediction.

schedule : dict:

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

scored_table_name : string

The name of the source table for this prediction.

id : integer

The ID of the prediction.

error : string

The error, if any, of the last run of this prediction.

model_id : integer

The ID of the model used for this prediction.

state : string

The state of the last run of this prediction.

finished_at : string/date-time

The end time of the last run of this prediction.

output_table_name : string

The name of the output table for this prediction.

scored_table_id : integer

The ID of the source table for this prediction.

primary_key : list

The primary key or composite keys of the table being predicted.

scored_tables : list:

```
An array of created prediction tables.
- score_stats : list::
    An array of metrics on the created predictions.
    - avg_score : number/float
        The average score.
    - min_score : number/float
        The minimum score.
    - max_score : number/float
        The maximum score.
    - histogram : list
        The histogram of the distribution of scores.
    - score_name : string
        The name of the score.
- name : string
    The name of table with created predictions.
- id : integer
    The ID of the table with created predictions.
- created_at : string/date-time
    The time when the table with created predictions was created.
- schema : string
    The schema of table with created predictions.
```

last_run : dict:

```
- started_at : string/time
    The time that the run started.
- id : integer
- error : string
    The error message for this run, if present.
```

```

- created_at : string/time
    The time that the run was queued.
- state : string
- finished_at : string/time
    The time that the run completed.

```

limiting_sql : string

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

post_runs (*id*)

Start a run

Parameters *id* : integer

The ID of the prediction.

Returns *prediction_id* : integer

The ID of the prediction.

score_stats : list:

```

An array of metrics on the created predictions.
- avg_score : number/float
    The average score.
- min_score : number/float
    The minimum score.
- max_score : number/float
    The maximum score.
- histogram : list
    The histogram of the distribution of scores.
- score_name : string
    The name of the score.

```

name : string

The name of table created by this predictions run.

state : string

The state of the prediction run.

exception : string

The exception, if any, returned by the prediction run.

created_at : string/date-time

The time when the table with created predictions was created.

id : integer

The ID of the prediction run.

put_schedules (*id*, ***kwargs*)

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_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

score_on_model_build : boolean, optional

Whether the prediction will run after a rebuild of the associated model.

Returns **schedule** : dict:

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

id : integer

ID of the prediction associated with this schedule.

score_on_model_build : boolean

Whether the prediction will run after a rebuild of the associated model.

Projects

class Projects (*session, 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(**kwargs)</code>	List projects
<code>list_shares(id)</code>	List users and groups permissioned on this object
<code>post(description, name, **kwargs)</code>	Create a project
<code>put(project_id, **kwargs)</code>	Update a project
<code>put_archive(id, status)</code>	Update the archive status of this object
<code>put_shares_groups(id, permission_level, ...)</code>	Set the permissions groups has on this object
<code>put_shares_users(id, user_ids, permission_level)</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 *archived* : string

The archival status of the requested object(s).

models : list:

```
- state : string
- updated_at : string/time
- id : integer
  The object ID.
- created_at : string/time
- name : string
```

tables : list:

```
- column_count : integer
- updated_at : string/time
- schema : string
- row_count : integer
- created_at : string/time
- name : string
```

description : string

A description of the project

author : dict:

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

updated_at : string/time

script_templates : list:

```
- updated_at : string/time
- id : integer
    The object ID.
- created_at : string/time
- name : string
```

all_objects : list:

```
- project_id : integer
- author : string
- archived : string
    The archival status of the requested object(s).
- name : string
- object_id : integer
- sub_type : string
- fco_type : string
- object_type : string
- icon : string
```

surveys : list:

```
- updated_at : string/time
- id : integer
    The object ID.
- created_at : string/time
```

imports : list:

```
- id : integer
    The object ID.
- updated_at : string/time
- type : string
- state : string
- finished_at : string/time
- created_at : string/time
- name : string
```

created_at : string/time

files : list:

```
- file_name : string
- file_size : integer
```

```
- updated_at : string/time
- id : integer
  The object ID.
- created_at : string/time
```

reports : list:

```
- state : string
- updated_at : string/time
- id : integer
  The object ID.
- created_at : string/time
- name : string
```

users : list:

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

hidden : boolean

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

auto_share : boolean

app_instances : list:

```
- updated_at : string/time
- slug : string
- id : integer
  The object ID.
- created_at : string/time
- name : string
```

note : string

id : integer

The ID for this project.

scripts : list:

```
- id : integer
  The object ID.
- updated_at : string/time
- type : string
- state : string
- finished_at : string/time
- created_at : string/time
- name : string
```

name : string

The name of this project.

list (***kwargs*)

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.

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 **archived** : string

The archival status of the requested object(s).

author : dict:

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

description : string

A description of the project

users : list:

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

name : string

The name of this project.

updated_at : string/time

auto_share : boolean

created_at : string/time

id : integer

The ID for this project.

list_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **total_group_shares** : integer

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

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
```

readers : 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.

post (*description*, *name*, ***kwargs*)

Create a project

Parameters **description** : string

A description of the project

name : string

The name of this project.

hidden : boolean, 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

note : string, optional

Notes for the project

Returns **archived** : string

The archival status of the requested object(s).

models : list:

```
- state : string
- updated_at : string/time
- id : integer
  The object ID.
- created_at : string/time
- name : string
```

tables : list:

```
- column_count : integer
- updated_at : string/time
- schema : string
- row_count : integer
- created_at : string/time
- name : string
```

description : string

A description of the project

author : dict:

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

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

updated_at : string/time

script_templates : list:

```
- updated_at : string/time
- id : integer
    The object ID.
- created_at : string/time
- name : string
```

all_objects : list:

```
- project_id : integer
- author : string
- archived : string
    The archival status of the requested object(s).
- name : string
- object_id : integer
- sub_type : string
- fco_type : string
- object_type : string
- icon : string
```

surveys : list:

```
- updated_at : string/time
- id : integer
    The object ID.
- created_at : string/time
```

imports : list:

```
- id : integer
    The object ID.
- updated_at : string/time
- type : string
- state : string
- finished_at : string/time
- created_at : string/time
- name : string
```

created_at : string/time

files : list:

```
- file_name : string
- file_size : integer
- updated_at : string/time
- id : integer
    The object ID.
- created_at : string/time
```

reports : list:

```
- state : string
- updated_at : string/time
- id : integer
    The object ID.
- created_at : string/time
- name : string
```

users : list:

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

hidden : boolean

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

auto_share : boolean

app_instances : list:

```
- updated_at : string/time
- slug : string
- id : integer
    The object ID.
- created_at : string/time
- name : string
```

note : string

id : integer

The ID for this project.

scripts : list:

```
- id : integer
    The object ID.
- updated_at : string/time
- type : string
- state : string
- finished_at : string/time
- created_at : string/time
- name : string
```

name : string

The name of this project.

put (*project_id*, ***kwargs*)
Update a project

Parameters `project_id` : integer

description : string, optional

A description of the project

name : string, optional

The name of this project.

note : string, optional

Notes for the project

Returns `archived` : string

The archival status of the requested object(s).

models : list:

```
- state : string
- updated_at : string/time
- id : integer
  The object ID.
- created_at : string/time
- name : string
```

tables : list:

```
- column_count : integer
- updated_at : string/time
- schema : string
- row_count : integer
- created_at : string/time
- name : string
```

description : string

A description of the project

author : dict:

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

updated_at : string/time

script_templates : list:

```
- updated_at : string/time
- id : integer
  The object ID.
- created_at : string/time
- name : string
```

all_objects : list:

```
- project_id : integer
- author : string
- archived : string
    The archival status of the requested object(s).
- name : string
- object_id : integer
- sub_type : string
- fco_type : string
- object_type : string
- icon : string
```

surveys : list:

```
- updated_at : string/time
- id : integer
    The object ID.
- created_at : string/time
```

imports : list:

```
- id : integer
    The object ID.
- updated_at : string/time
- type : string
- state : string
- finished_at : string/time
- created_at : string/time
- name : string
```

created_at : string/time**files** : list:

```
- file_name : string
- file_size : integer
- updated_at : string/time
- id : integer
    The object ID.
- created_at : string/time
```

reports : list:

```
- state : string
- updated_at : string/time
- id : integer
    The object ID.
- created_at : string/time
- name : string
```

users : list:

```
Users who can see the project
- username : string
    This user's username.
- initials : string
    This user's initials.
- online : boolean
```

```

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

```

hidden : boolean

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

auto_share : boolean

app_instances : list:

```

- updated_at : string/time
- slug : string
- id : integer
    The object ID.
- created_at : string/time
- name : string

```

note : string

id : integer

The ID for this project.

scripts : list:

```

- id : integer
    The object ID.
- updated_at : string/time
- type : string
- state : string
- finished_at : string/time
- created_at : string/time
- name : string

```

name : string

The name of this project.

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 **archived** : string

The archival status of the requested object(s).

models : list:

```

- state : string
- updated_at : string/time
- id : integer

```

```
The object ID.
- created_at : string/time
- name : string
```

tables : list:

```
- column_count : integer
- updated_at : string/time
- schema : string
- row_count : integer
- created_at : string/time
- name : string
```

description : string

A description of the project

author : dict:

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

updated_at : string/time

script_templates : list:

```
- updated_at : string/time
- id : integer
    The object ID.
- created_at : string/time
- name : string
```

all_objects : list:

```
- project_id : integer
- author : string
- archived : string
    The archival status of the requested object(s).
- name : string
- object_id : integer
- sub_type : string
- fco_type : string
- object_type : string
- icon : string
```

surveys : list:

```
- updated_at : string/time
- id : integer
    The object ID.
- created_at : string/time
```


imports : list:

```
- id : integer
    The object ID.
- updated_at : string/time
- type : string
- state : string
- finished_at : string/time
- created_at : string/time
- name : string
```

created_at : string/time**files** : list:

```
- file_name : string
- file_size : integer
- updated_at : string/time
- id : integer
    The object ID.
- created_at : string/time
```

reports : list:

```
- state : string
- updated_at : string/time
- id : integer
    The object ID.
- created_at : string/time
- name : string
```

users : list:

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

hidden : boolean

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

auto_share : boolean**app_instances** : list:

```
- updated_at : string/time
- slug : string
- id : integer
    The object ID.
- created_at : string/time
- name : string
```

note : string

id : integer

The ID for this project.

scripts : list:

```
- id : integer
    The object ID.
- updated_at : string/time
- type : string
- state : string
- finished_at : string/time
- created_at : string/time
- name : string
```

name : string

The name of this project.

put_shares_groups (*id*, *permission_level*, *group_ids*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

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

group_ids : list

An array of one or more group IDs

Returns **total_group_shares** : integer

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

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
```

readers : 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.

put_shares_users (*id*, *user_ids*, *permission_level*)

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”

Returns **total_group_shares** : integer

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

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

```

readers : 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.

Queries

class **Queries** (*session*, *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(**kwargs)</code>	List all queries
<code>list_runs(id, **kwargs)</code>	List runs for the given query
<code>post(preview_rows, database, sql, **kwargs)</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 **credential** : integer

The credential ID.

started_at : string/date-time

The start time of the last run.

result_rows : list

A preview of rows returned by the query.

id : integer

The query ID.

updated_at : string/time

state : string

The state of the last run.

database : integer

The database ID.

sql : string

The SQL to execute.

name : string

The name of the query.

result_columns : list

A preview of columns returned by the query.

finished_at : string/date-time

The end time of the last run.

exception : string

Exception returned from the query, null if the query was a success.

author : dict:

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

created_at : string/time

hidden : boolean

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

report_id : integer

The ID of the report associated with this query.

script_id : integer

The ID of the script associated with this query.

last_run_id : integer

The ID of the last run.

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 **started_at** : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

finished_at : string/time

The time the last run completed.

query_id : integer

The ID of the query.

list (***kwargs*)

List all queries

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.

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 credential : integer

The credential ID.

started_at : string/date-time

The start time of the last run.

result_rows : list

A preview of rows returned by the query.

id : integer

The query ID.

updated_at : string/time

state : string

The state of the last run.

database : integer

The database ID.

sql : string

The SQL to execute.

result_columns : list

A preview of columns returned by the query.

finished_at : string/date-time

The end time of the last run.

exception : string

Exception returned from the query, null if the query was a success.

created_at : string/time

report_id : integer

The ID of the report associated with this query.

script_id : integer

The ID of the script associated with this query.

preview_rows : integer

The number of rows to save from the query's result (maximum: 100).

last_run_id : integer

The ID of the last run.

list_runs (*id*, ***kwargs*)

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 started_at : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

finished_at : string/time

The time the last run completed.

query_id : integer

The ID of the query.

post (*preview_rows*, *database*, *sql*, ***kwargs*)
Execute a query

Parameters preview_rows : integer

The number of rows to save from the query's result (maximum: 100).

database : integer

The database ID.

sql : string

The SQL to execute.

credential : integer, optional

The credential ID.

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.

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].

filename_prefix : string, optional

The output filename prefix.

hidden : boolean, 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

interactive : boolean, optional

Deprecated and not used.

Returns started_at : string/date-time

The start time of the last run.

id : integer

The query ID.

filename_prefix : string

The output filename prefix.

state : string

The state of the last run.

database : integer

The database ID.

sql : string

The SQL to execute.

finished_at : string/date-time

The end time of the last run.

last_run_id : integer

The ID of the last run.

hidden : boolean

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

compression : string

The type of compression. One of gzip or zip, or none [default: gzip].

created_at : string/time

report_id : integer

The ID of the report associated with this query.

exception : string

Exception returned from the query, null if the query was a success.

preview_rows : integer

The number of rows to save from the query's result (maximum: 100).

credential : integer

The credential ID.

column_delimiter : string

The delimiter to use. One of comma or tab, or pipe [default: comma].

result_rows : list

A preview of rows returned by the query.

script_id : integer

The ID of the script associated with this query.

include_header : boolean

Whether the CSV output should include a header row [default: true].

updated_at : string/time

unquoted : boolean

If true, will not quote fields.

result_columns : list

A preview of columns returned by the query.

interactive : boolean

Deprecated and not used.

post_runs (*id*)

Start a run

Parameters id : integer

The ID of the query.

Returns started_at : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

finished_at : string/time

The time the last run completed.

query_id : integer

The ID of the query.

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 credential : integer

The credential ID.

started_at : string/date-time

The start time of the last run.

result_rows : list

A preview of rows returned by the query.

id : integer

The query ID.

updated_at : string/time

state : string

The state of the last run.

database : integer

The database ID.

sql : string

The SQL to execute.

name : string

The name of the query.

result_columns : list

A preview of columns returned by the query.

finished_at : string/date-time

The end time of the last run.

exception : string

Exception returned from the query, null if the query was a success.

author : dict:

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

created_at : string/time

hidden : boolean

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

report_id : integer

The ID of the report associated with this query.

script_id : integer

The ID of the script associated with this query.

last_run_id : integer

The ID of the last run.

Reports

class Reports (*session*, *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>get</i> (id)	Show a single report
<i>list</i> (**kwargs)	List the reports visible to the current user
<i>list_projects</i> (id)	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>patch</i> (id, **kwargs)	Update a report
<i>patch_snapshots</i> (id, **kwargs)	Update the report's snapshot automation settings
<i>post</i> (**kwargs)	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, **kwargs)	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, permission_level, ...)	Set the permissions groups has on this object
<i>put_shares_users</i> (id, user_ids, permission_level)	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

get (*id*)

Show a single report

Parameters **id** : integer

The ID of this report.

Returns **job_path** : string

The link to details of the job that backs this report.

id : integer

The ID of this report.

state : string

The status of the report's last run.

viz_updated_at : string/time

The time that the report's visualization was last updated.

finished_at : string/time

The time that the report's last run finished.

script : dict:

```
- sql : string
    The raw SQL query for the script.
- id : integer
    The ID for the script.
- name : string
    The name of the script.
```

archived : string

The archival status of the requested object(s).

created_at : string/time

user : dict:

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

hidden : boolean

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

api_key : string

A Civis API key that can be used by this report.

config : string

Any configuration metadata for this report.

last_run : dict:

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

auth_code_url : string

app_state : dict

Any application state blob for this report.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

name : string

The name of the report.

updated_at : string/time

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

tableau_id : integer

projects : list:

A `list` of projects containing the report.

- `id` : integer
The ID `for` the project.
- `name` : string
The name of the project.

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

auth_data_url : string

list (***kwargs*)

List the reports visible to the current user

Parameters **type** : string, optional

If specified, return report of these types. It accepts a comma-separated list, possible values are 'tableau', '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.

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 **job_path** : string

The link to details of the job that backs this report.

id : integer

The ID of this report.

updated_at : string/time

state : string

The status of the report's last run.

template_id : integer

The ID of the template used for this report.

viz_updated_at : string/time

The time that the report's visualization was last updated.

finished_at : string/time

The time that the report's last run finished.

script : dict:

```
- sql : string
    The raw SQL query for the script.
- id : integer
    The ID for the script.
- name : string
    The name of the script.
```

auth_thumbnail_url : string

URL for a thumbnail of the report.

tableau_id : integer

archived : string

The archival status of the requested object(s).

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```


created_at : string/time

user : dict:

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

last_run : dict:

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

name : string

The name of the report.

list_projects (*id*)

List the projects a Report belongs to

Parameters **id** : integer

The ID of the resource.

Returns **archived** : string

The archival status of the requested object(s).

author : dict:

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

description : string

A description of the project

users : list:

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

name : string

The name of this project.

updated_at : string/time

auto_share : boolean

created_at : string/time

id : integer

The ID for this project.

list_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **total_group_shares** : integer

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

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
```

readers : 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.

list_snapshots (*id*)

Get details about the report's snapshot automation settings

Parameters *id* : integer

The ID of this report.

Returns *schedule* : dict:

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

email_template : string

Custom email template.

send_email_on_completion : boolean

Whether the job will send emails on completion.

width : integer

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

id : integer

The ID of this report.

recipient_email_addresses : string

Email addresses to send report to, comma separated.

parent_id : integer

The ID of the parent job that will trigger this snapshot.

state : string

The status of the job's last run.

height : integer

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

finished_at : string/time

The time that the job's last run finished.

email_subject : string

Subject for Email.

patch (*id*, ***kwargs*)

Update a report

Parameters **id** : integer

The ID of the report to modify.

app_state : dict, optional

The application state blob for this report.

name : string, optional

The name of the report.

provide_api_key : boolean, optional

Allow the report to provide an API key to front-end code.

code_body : string, optional

The code for the report visualization.

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.

script_id : integer, optional

The ID of the job (a script or a query) used to create this report.

config : string, optional

Returns **job_path** : string

The link to details of the job that backs this report.

id : integer

The ID of this report.

state : string

The status of the report's last run.

viz_updated_at : string/time

The time that the report's visualization was last updated.

finished_at : string/time

The time that the report's last run finished.

script : dict:

```
- sql : string
    The raw SQL query for the script.
- id : integer
    The ID for the script.
- name : string
    The name of the script.
```

archived : string

The archival status of the requested object(s).

created_at : string/time

user : dict:

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

hidden : boolean

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

api_key : string

A Civis API key that can be used by this report.

config : string

Any configuration metadata for this report.

last_run : dict:

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

auth_code_url : string

app_state : dict

Any application state blob for this report.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

name : string

The name of the report.

updated_at : string/time

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

tableau_id : integer

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

auth_data_url : string

patch_snapshots (*id*, ***kwargs*)

Update the report's snapshot automation settings

Parameters id : integer

The ID of this report.

schedule : dict, optional:

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

email_template : string, optional

Custom email template.

send_email_on_completion : boolean, optional

Whether the job will send emails on completion.

width : integer, optional

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

recipient_email_addresses : string, optional

Email addresses to send report to, comma separated.

parent_id : integer, optional

The ID of the parent job that will trigger this snapshot.

state : string, optional

The status of the job's last run.

height : integer, optional

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

finished_at : string/time, optional

The time that the job's last run finished.

email_subject : string, optional

Subject for Email.

Returns **schedule** : dict:

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

email_template : string

Custom email template.

send_email_on_completion : boolean

Whether the job will send emails on completion.

width : integer

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

id : integer

The ID of this report.

recipient_email_addresses : string

Email addresses to send report to, comma separated.

parent_id : integer

The ID of the parent job that will trigger this snapshot.

state : string

The status of the job's last run.

height : integer

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

finished_at : string/time

The time that the job's last run finished.

email_subject : string

Subject for Email.

post (***kwargs*)

Create a report

Parameters **app_state** : dict, optional

Any application state blob for this report.

hidden : boolean, 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

name : string, optional

The name of the report.

provide_api_key : boolean, optional

Allow the report to provide an API key to front-end code.

code_body : string, optional

The code for the report visualization.

template_id : integer, optional

The ID of the template used for this report.

script_id : integer, optional

The ID of the job (a script or a query) used to create this report.

Returns **job_path** : string

The link to details of the job that backs this report.

id : integer

The ID of this report.

state : string

The status of the report's last run.

viz_updated_at : string/time

The time that the report's visualization was last updated.

finished_at : string/time

The time that the report's last run finished.

script : dict:

```
- sql : string
    The raw SQL query for the script.
- id : integer
    The ID for the script.
- name : string
    The name of the script.
```

archived : string

The archival status of the requested object(s).

created_at : string/time

user : dict:

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

hidden : boolean

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

api_key : string

A Civis API key that can be used by this report.

config : string

Any configuration metadata for this report.

last_run : dict:

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

auth_code_url : string

app_state : dict

Any application state blob for this report.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

name : string

The name of the report.

updated_at : string/time

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

tableau_id : integer

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

auth_data_url : string

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 **job_path** : string

The link to details of the job that backs this report.

id : integer

The ID of this report.

state : string

The status of the report's last run.

viz_updated_at : string/time

The time that the report's visualization was last updated.

finished_at : string/time

The time that the report's last run finished.

script : dict:

```
- sql : string
    The raw SQL query for the script.
- id : integer
    The ID for the script.
- name : string
    The name of the script.
```

archived : string

The archival status of the requested object(s).

created_at : string/time

user : dict:

```

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

```

hidden : boolean

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

api_key : string

A Civis API key that can be used by this report.

config : string

Any configuration metadata for this report.

last_run : dict:

```

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

```

auth_code_url : string

app_state : dict

Any application state blob for this report.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

name : string

The name of the report.

updated_at : string/time

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

tableau_id : integer

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

auth_data_url : string

post_snapshots (*id*, ***kwargs*)

Generate and optionally email a snapshot of the specified report

Parameters **id** : integer

The ID of this report.

schedule : dict, optional:

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

email_template : string, optional

Custom email template.

send_email_on_completion : boolean, optional

Whether the job will send emails on completion.

width : integer, optional

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

recipient_email_addresses : string, optional

Email addresses to send report to, comma separated.

parent_id : integer, optional

The ID of the parent job that will trigger this snapshot.

state : string, optional

The status of the job's last run.

height : integer, optional

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

finished_at : string/time, optional

The time that the job's last run finished.

email_subject : string, optional

Subject for Email.

Returns schedule : dict:

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

email_template : string

Custom email template.

send_email_on_completion : boolean

Whether the job will send emails on completion.

width : integer

The width of the cropped snapshot image in screen pixels. The default value is 1440 pixels. Minimum value is 600 pixels.

id : integer

The ID of this report.

recipient_email_addresses : string

Email addresses to send report to, comma separated.

parent_id : integer

The ID of the parent job that will trigger this snapshot.

state : string

The status of the job's last run.

height : integer

The height of the cropped snapshot image in screen pixels. The default value is 900 pixels. Minimum value is 600 pixels.

finished_at : string/time

The time that the job's last run finished.

email_subject : string

Subject for Email.

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 **job_path** : string

The link to details of the job that backs this report.

id : integer

The ID of this report.

state : string

The status of the report's last run.

viz_updated_at : string/time

The time that the report's visualization was last updated.

finished_at : string/time

The time that the report's last run finished.

script : dict:

```
- sql : string
    The raw SQL query for the script.
- id : integer
    The ID for the script.
- name : string
    The name of the script.
```

archived : string

The archival status of the requested object(s).

created_at : string/time

user : dict:

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

hidden : boolean

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

api_key : string

A Civis API key that can be used by this report.

config : string

Any configuration metadata for this report.

last_run : dict:

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

auth_code_url : string

app_state : dict

Any application state blob for this report.

provide_api_key : boolean

Whether the report requests an API Key from the report viewer.

name : string

The name of the report.

updated_at : string/time

api_key_id : integer

The ID of the API key. Can be used for auditing API use by this report.

template_id : integer

The ID of the template used for this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

tableau_id : integer

projects : list:

```
A list of projects containing the report.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

valid_output_file : boolean

Whether the job (a script or a query) that backs the report currently has a valid output file.

auth_data_url : string

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, permission_level, group_ids*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

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

group_ids : list

An array of one or more group IDs

Returns **total_group_shares** : integer

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

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
```

readers : 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.

put_shares_users (*id, user_ids, permission_level*)

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”

Returns **total_group_shares** : integer

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

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
```

readers : 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.

Scripts

class Scripts (*session, 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
<code>delete_sql_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>get(id)</code>	Get details about a script
<code>get_containers(id)</code>	View a container
<code>get_containers_runs(id, run_id)</code>	Check status of a run
<code>get_custom(id)</code>	Get a CustomScript
<code>get_custom_runs(id, run_id)</code>	Check status of a run
<code>get_javascript(id)</code>	Get a JavaScript Script
<code>get_javascript_runs(id, run_id)</code>	Check status of a run
<code>get_python3(id)</code>	Get a Python Script
<code>get_python3_runs(id, run_id)</code>	Check status of a run
<code>get_r(id)</code>	Get an R Script
<code>get_r_runs(id, run_id)</code>	Check status of a run
<code>get_sql(id)</code>	Get a SQL script
<code>get_sql_runs(id, run_id)</code>	Check status of a run
<code>list(**kwargs)</code>	List scripts
<code>list_containers_projects(id)</code>	List the projects a container docker belongs to
<code>list_containers_runs(id, **kwargs)</code>	List runs for the given container
<code>list_containers_runs_logs(id, run_id, **kwargs)</code>	Get the logs for a run
<code>list_containers_runs_outputs(id, run_id, ...)</code>	List the outputs for a run
<code>list_containers_shares(id)</code>	List users and groups permissioned on this object

Continued on next page

Table 5.14 – continued from previous page

<i>list_custom(**kwargs)</i>	List Custom Scripts
<i>list_custom_projects(id)</i>	List the projects a Job belongs to
<i>list_custom_runs(id, **kwargs)</i>	List runs for the given custom
<i>list_custom_runs_logs(id, run_id, **kwargs)</i>	Get the logs for a run
<i>list_custom_runs_outputs(id, run_id, **kwargs)</i>	List the outputs for a run
<i>list_custom_shares(id)</i>	List users and groups permissioned on this object
<i>list_history(id)</i>	Get the run history and outputs of this script
<i>list_javascript_projects(id)</i>	List the projects a scripted sql belongs to
<i>list_javascript_runs(id, **kwargs)</i>	List runs for the given javascript
<i>list_javascript_runs_logs(id, run_id, **kwargs)</i>	Get the logs for a run
<i>list_javascript_runs_outputs(id, run_id, ...)</i>	List the outputs for a run
<i>list_javascript_shares(id)</i>	List users and groups permissioned on this object
<i>list_python3_projects(id)</i>	List the projects a python docker belongs to
<i>list_python3_runs(id, **kwargs)</i>	List runs for the given python
<i>list_python3_runs_logs(id, run_id, **kwargs)</i>	Get the logs for a run
<i>list_python3_runs_outputs(id, run_id, **kwargs)</i>	List the outputs for a run
<i>list_python3_shares(id)</i>	List users and groups permissioned on this object
<i>list_r_projects(id)</i>	List the projects a r docker belongs to
<i>list_r_runs(id, **kwargs)</i>	List runs for the given r
<i>list_r_runs_logs(id, run_id, **kwargs)</i>	Get the logs for a run
<i>list_r_runs_outputs(id, run_id, **kwargs)</i>	List the outputs for a run
<i>list_r_shares(id)</i>	List users and groups permissioned on this object
<i>list_sql_projects(id)</i>	List the projects a scripts belongs to
<i>list_sql_runs(id, **kwargs)</i>	List runs for the given sql
<i>list_sql_runs_logs(id, run_id, **kwargs)</i>	Get the logs for a run
<i>list_sql_runs_outputs(id, run_id, **kwargs)</i>	List the outputs for a run
<i>list_sql_shares(id)</i>	List users and groups permissioned on this object
<i>list_types()</i>	List available script types
<i>patch(id, **kwargs)</i>	Update a script
<i>patch_containers(id, **kwargs)</i>	Update a container
<i>patch_containers_runs(id, run_id, **kwargs)</i>	Update a run
<i>patch_custom(id, **kwargs)</i>	Update some attributes of this CustomScript
<i>patch_javascript(id, **kwargs)</i>	Update some attributes of this JavaScript Script
<i>patch_python3(id, **kwargs)</i>	Update some attributes of this Python Script
<i>patch_r(id, **kwargs)</i>	Update some attributes of this R Script
<i>patch_sql(id, **kwargs)</i>	Update some attributes of this SQL script
<i>post(name, remote_host_id, sql, ...)</i>	Create a script
<i>post_cancel(id)</i>	Cancel a run
<i>post_containers(required_resources, ...)</i>	Create a container
<i>post_containers_runs(id)</i>	Start a run
<i>post_containers_runs_heartbeats(id, run_id)</i>	Indicate that the given run is being handled
<i>post_containers_runs_logs(id, run_id, **kwargs)</i>	Add log messages
<i>post_containers_runs_outputs(id, run_id, ...)</i>	Add an output for a run

Continued on next page

Table 5.14 – continued from previous page

<code>post_custom(from_template_id, **kwargs)</code>	Create a CustomScript
<code>post_custom_runs(id)</code>	Start a run
<code>post_custom_runs_outputs(id, run_id, ...)</code>	Add an output for a run
<code>post_javascript(name, remote_host_id, ...)</code>	Create a JavaScript Script
<code>post_javascript_runs(id)</code>	Start a run
<code>post_javascript_runs_outputs(id, run_id, ...)</code>	Add an output for a run
<code>post_python3(name, source, **kwargs)</code>	Create a Python Script
<code>post_python3_runs(id)</code>	Start a run
<code>post_python3_runs_outputs(id, run_id, ...)</code>	Add an output for a run
<code>post_r(name, source, **kwargs)</code>	Create an R Script
<code>post_r_runs(id)</code>	Start a run
<code>post_r_runs_outputs(id, run_id, object_type, ...)</code>	Add an output for a run
<code>post_run(id)</code>	Run a script
<code>post_sql(name, remote_host_id, sql, ...)</code>	Create a SQL script
<code>post_sql_runs(id)</code>	Start a run
<code>put_containers(id, required_resources, ...)</code>	Edit a container
<code>put_containers_archive(id, status)</code>	Update the archive status of this object
<code>put_containers_projects(id, project_id)</code>	Add a container docker to a project
<code>put_containers_shares_groups(id, ...)</code>	Set the permissions groups has on this object
<code>put_containers_shares_users(id, user_ids, ...)</code>	Set the permissions users have on this object
<code>put_custom(id, **kwargs)</code>	Replace all attributes of this CustomScript
<code>put_custom_archive(id, status)</code>	Update the archive status of this object
<code>put_custom_projects(id, project_id)</code>	Add a Job to a project
<code>put_custom_shares_groups(id, ...)</code>	Set the permissions groups has on this object
<code>put_custom_shares_users(id, user_ids, ...)</code>	Set the permissions users have on this object
<code>put_javascript(id, name, remote_host_id, ...)</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, ...)</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, **kwargs)</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, ...)</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, **kwargs)</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, permission_level, ...)</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, remote_host_id, sql, ...)</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, permission_level, ...)</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 *schedule* : dict:

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

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

sql : string

The raw SQL query for the script.

finished_at : string/time

The time that the script's last run finished.

params : list:

A definition of the parameters this script accepts **in** the arguments_↵
 ↵field.

- required : boolean
 Whether this param **is** required.
- description : string
 A short sentence **or** fragment describing this parameter to the_↵
 ↵end user.
- name : string
 The variable's name as used within your code.
- value : string
 The value you would like to **set** this param to. Setting this_↵
 ↵value makes
 this parameter a fixed param.
- type : string
 The **type** of parameter. Valid options: string, integer, float,_↵
 ↵bool,
 file, database, credential_aws, credential_redshift, **or**
 credential_custom
- label : string
 The label to present to users when asking them **for** the value.
- 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.

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the script.

template_script_id : integer

The ID of the template script, if any.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

updated_at : string/time

The time this script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

type : string

The type of script.

next_run_at : string/time

The time of the next scheduled run.

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

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

get_containers (*id*)

View a container

Parameters **id** : integer

The ID for the script.

Returns **schedule** : dict:

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

repo_ref : string

The tag or branch of the github repo to clone into the container.

hidden : boolean

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

author : dict:

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

repo_http_uri : string

The location of a github repo to clone into the container, e.g. `github.com/my-user/my-repo.git`.

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.

```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

```

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

```

name : string

The name of the container.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template script.

type : string

The type of the script (e.g Container)

time_zone : string

The time zone of this script.

docker_image_name : string

The name of the docker image to pull from DockerHub.

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.
```

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

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.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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



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

remote_host_credential_id : integer

The id of the database credentials to pass into the environment of the container.

id : integer

The ID for the script.

notifications : dict:

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

docker_command : string

The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]

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 **started_at** : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

finished_at : string/time

The time the last run completed.

container_id : integer

The ID of the container.

get_custom (*id*)

Get a CustomScript

Parameters **id** : integer

Returns **schedule** : dict:

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

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

params : list:

A definition of the parameters this script accepts **in** the arguments_↵
 ↵field.

- required : boolean
 Whether this param **is** required.
- description : string
 A short sentence **or** fragment describing this parameter to the_↵
 ↵end user.
- name : string
 The variable's name as used within your code.
- value : string
 The value you would like to **set** this param to. Setting this_↵
 ↵value makes
 this parameter a fixed param.
- type : string
 The **type** of parameter. Valid options: string, integer, **float**,_↵
 ↵bool,
 file, database, credential_aws, credential_redshift, **or**
 credential_custom
- label : string
 The label to present to users when asking them **for** the value.
- 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.

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

last_run : dict:

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

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template script.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g Custom)

time_zone : string

The time zone of this script.

target_project_id : integer

Target project to which script outputs will be added.

code_preview : string

The code that this script will run with arguments inserted.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on
- urls : list
```

```

    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.

```

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 *started_at* : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

custom_id : integer

The ID of the custom.

finished_at : string/time

The time the last run completed.

get_javascript (*id*)

Get a JavaScript Script

Parameters *id* : integer**Returns** *schedule* : dict:

```

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

```

```
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,_
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
```

```

- 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.

```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

```

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

```

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on
- urls : list
    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
```



```

- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
    successfully.
- success_email_subject : string
    Custom subject line for success e-mail.

```

source : string

The body/text of the script.

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 *started_at* : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

javascript_id : integer

The ID of the javascript.

finished_at : string/time

The time the last run completed.

get_python3 (*id*)

Get a Python Script

Parameters *id* : integer

Returns *schedule* : dict:

```

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

```

```
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↳value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↳bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.

```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

```

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

```

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.

```

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

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

source : string

The body/text of the script.

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 *started_at* : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

finished_at : string/time

The time the last run completed.

python_id : integer

The ID of the python.

get_r (*id*)

Get an R Script

Parameters *id* : integer

Returns `schedule` : dict:

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

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↪bool,
```

```

    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.

```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

```

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

```

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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



```

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

```

id : integer

The ID for the script.

notifications : dict:

```

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

```

source : string

The body/text of the script.

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 **started_at** : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

r_id : integer

The ID of the r.

finished_at : string/time

The time the last run completed.

get_sql (*id*)

Get a SQL script

Parameters *id* : integer

Returns *schedule* : dict:

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

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

sql : string

The raw SQL query for the script.

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

csv_settings : dict:

```
- force_multifile : boolean
    Whether or not the csv should be split into multiple files.
↪Default:
    false
- column_delimiter : string
    Which delimiter to use, one of "comma", "tab", or "pipe".
↪Default:
    comma
- 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
- filename_prefix : string
    A user specified filename prefix for the output file to have.
↪Default:
    null
- unquoted : boolean
    Whether or not to quote fields. Default: false
```

params : list:

```
A definition of the parameters this script accepts in the arguments.
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

code_preview : string

The code that this script will run with arguments inserted.

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

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

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 *started_at* : string/time

The time the last run started.

id : integer

The ID of this run.

error : string

The error message for this run, if present.

is_cancel_requested : boolean

True if run cancel requested, else false.

sql_id : integer

The ID of this sql.

state : string

The state of this run.

output : list:

A `list` of the outputs of this script.

- `path` : string
The temporary link to download this output file, valid `for 36_`
`↪hours`.
- `file_id` : integer
The unique ID of the output file.
- `output_name` : string
The name of the output file.

finished_at : string/time

The time that this run finished.

list (***kwargs*)

List scripts

Parameters *type* : string, optional

If specified, return objects of these types. The valid types are 'sql', 'python3', 'r', and 'javascript'.

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 from_template_id : integer

The ID of the template this script uses, if any.

author : dict:

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

updated_at : string/time

The time the script was last updated.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

state : string

The status of the script's last run.

name : string

The name of the script.

finished_at : string/time

The time that the script's last run finished.

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.
```

is_template : boolean

Whether others scripts use this one as a template.

archived : string

The archival status of the requested object(s).

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

parent_id : integer

The ID of the parent job that will trigger this script

template_script_id : integer

The ID of the template script, if any.

last_run : dict:

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

id : integer

The ID for the script.

list_containers_projects (*id*)

List the projects a container docker belongs to

Parameters id : integer

The ID of the resource.

Returns archived : string

The archival status of the requested object(s).

author : dict:

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

description : string

A description of the project

users : list:

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

name : string

The name of this project.

updated_at : string/time

auto_share : boolean

created_at : string/time

id : integer

The ID for this project.

list_containers_runs (*id*, ***kwargs*)

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 started_at : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

finished_at : string/time

The time the last run completed.

container_id : integer

The ID of the container.

list_containers_runs_logs (*id*, *run_id*, ***kwargs*)

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 level : string

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

message : string

The log message.

id : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

list_containers_runs_outputs (*id*, *run_id*, ***kwargs*)

List the outputs for a run

Parameters **id** : integer

The ID of the output.

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 **link** : string

The link to retrieve the output object.

name : string

The name of the output object.

object_type : string

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

object_id : integer

The ID of the output object.

list_containers_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **total_group_shares** : integer

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

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
```

readers : 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.

list_custom (**kwargs)
List Custom Scripts

Parameters from_template_id : integer, optional

The template script that this app uses.

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 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 **author** : dict:

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

updated_at : string/time

The time the script was last updated.

type : string

The type of the script (e.g Custom)

state : string

The status of the script's last run.

from_template_id : integer

The ID of the template script.

finished_at : string/time

The time that the script's last run finished.

name : string

The name of the script.

archived : string

The archival status of the requested object(s).

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

parent_id : integer

The ID of the parent job that will trigger this script

last_run : dict:

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

id : integer

The ID for the script.

list_custom_projects (*id*)

List the projects a Job belongs to

Parameters **id** : integer

The ID of the resource.

Returns **archived** : string

The archival status of the requested object(s).

author : dict:

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

description : string

A description of the project

users : list:

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

```

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

```

name : string

The name of this project.

updated_at : string/time

auto_share : boolean

created_at : string/time

id : integer

The ID for this project.

list_custom_runs (*id*, ***kwargs*)

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 **started_at** : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

custom_id : integer

The ID of the custom.

finished_at : string/time

The time the last run completed.

list_custom_runs_logs (*id*, *run_id*, ***kwargs*)

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 **level** : string

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

message : string

The log message.

id : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

list_custom_runs_outputs (*id*, *run_id*, ***kwargs*)

List the outputs for a run

Parameters **id** : integer

The ID of the output.

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 link : string

The link to retrieve the output object.

name : string

The name of the output object.

object_type : string

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

object_id : integer

The ID of the output object.

list_custom_shares (*id*)

List users and groups permissioned on this object

Parameters id : integer

The ID of the object.

Returns total_group_shares : integer

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

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
```

readers : 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.

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.

error : string

The error message for this run, if present.

is_cancel_requested : boolean

True if run cancel requested, else false.

sql_id : integer

The ID of this sql.

state : string

The state of this run.

output : list:

```
A list of the outputs of this script.
- path : string
  The temporary link to download this output file, valid for 36
  ↪hours.
- file_id : integer
  The unique ID of the output file.
- output_name : string
  The name of the output file.
```

finished_at : string/time

The time that this run finished.

list_javascript_projects (*id*)

List the projects a scripted sql belongs to

Parameters **id** : integer

The ID of the resource.

Returns **archived** : string

The archival status of the requested object(s).

author : dict:

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

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

description : string

A description of the project

users : list:

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

name : string

The name of this project.

updated_at : string/time

auto_share : boolean

created_at : string/time

id : integer

The ID for this project.

list_javascript_runs (*id*, ***kwargs*)

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 **started_at** : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

javascript_id : integer

The ID of the javascript.

finished_at : string/time

The time the last run completed.

list_javascript_runs_logs (*id*, *run_id*, ***kwargs*)

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 **level** : string

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

message : string

The log message.

id : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

list_javascript_runs_outputs (*id*, *run_id*, ***kwargs*)

List the outputs for a run

Parameters *id* : integer

The ID of the output.

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 *link* : string

The link to retrieve the output object.

name : string

The name of the output object.

object_type : string

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

object_id : integer

The ID of the output object.

list_javascript_shares (*id*)

List users and groups permissioned on this object

Parameters *id* : integer

The ID of the object.

Returns *total_group_shares* : integer

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

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
```

readers : 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.

list_python3_projects (*id*)

List the projects a python docker belongs to

Parameters *id* : integer

The ID of the resource.

Returns *archived* : string

The archival status of the requested object(s).

author : dict:

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

description : string

A description of the project

users : list:

```

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

```

name : string

The name of this project.

updated_at : string/time

auto_share : boolean

created_at : string/time

id : integer

The ID for this project.

list_python3_runs (*id*, ***kwargs*)

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 **started_at** : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

finished_at : string/time

The time the last run completed.

python_id : integer

The ID of the python.

list_python3_runs_logs (*id, run_id, **kwargs*)

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 **level** : string

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

message : string

The log message.

id : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

list_python3_runs_outputs (*id, run_id, **kwargs*)

List the outputs for a run

Parameters **id** : integer

The ID of the output.

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 link : string

The link to retrieve the output object.

name : string

The name of the output object.

object_type : string

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

object_id : integer

The ID of the output object.

list_python3_shares (*id*)

List users and groups permissioned on this object

Parameters id : integer

The ID of the object.

Returns total_group_shares : integer

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

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
```

readers : 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.

list_r_projects (*id*)

List the projects a r docker belongs to

Parameters *id* : integer

The ID of the resource.

Returns *archived* : string

The archival status of the requested object(s).

author : dict:

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

description : string

A description of the project

users : list:

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

name : string

The name of this project.

updated_at : string/time

auto_share : boolean

created_at : string/time

id : integer

The ID for this project.

list_r_runs (*id*, ***kwargs*)

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 **started_at** : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

r_id : integer

The ID of the r.

finished_at : string/time

The time the last run completed.

list_r_runs_logs (*id*, *run_id*, ***kwargs*)

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 level : string

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

message : string

The log message.

id : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

list_r_runs_outputs (*id*, *run_id*, ***kwargs*)

List the outputs for a run

Parameters id : integer

The ID of the output.

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 link : string

The link to retrieve the output object.

name : string

The name of the output object.

object_type : string

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

object_id : integer

The ID of the output object.

list_r_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **total_group_shares** : integer

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

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
```

readers : 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.

list_sql_projects (*id*)

List the projects a scripts belongs to

Parameters **id** : integer

The ID of the resource.

Returns **archived** : string

The archival status of the requested object(s).

author : dict:

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

description : string

A description of the project

users : list:

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

name : string

The name of this project.

updated_at : string/time

auto_share : boolean

created_at : string/time

id : integer

The ID for this project.

list_sql_runs (*id*, ****kwargs**)

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 **started_at** : string/time

The time the last run started.

id : integer

The ID of this run.

error : string

The error message for this run, if present.

is_cancel_requested : boolean

True if run cancel requested, else false.

sql_id : integer

The ID of this sql.

state : string

The state of this run.

output : list:

```
A list of the outputs of this script.
- path : string
  The temporary link to download this output file, valid for 36
  ↪hours.
- file_id : integer
  The unique ID of the output file.
- output_name : string
  The name of the output file.
```

finished_at : string/time

The time that this run finished.

list_sql_runs_logs (*id*, *run_id*, ***kwargs*)

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 level : string

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

message : string

The log message.

id : integer

The ID of the log.

created_at : string/date-time

The time the log was created.

list_sql_runs_outputs (*id*, *run_id*, ***kwargs*)

List the outputs for a run

Parameters id : integer

The ID of the output.

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 link : string

The link to retrieve the output object.

name : string

The name of the output object.

object_type : string

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

object_id : integer

The ID of the output object.

list_sql_shares (*id*)

List users and groups permissioned on this object

Parameters *id* : integer

The ID of the object.

Returns *total_group_shares* : integer

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

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
```

readers : 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.

list_types ()

List available script types

Returns *name* : string

The name of the type.

patch (*id*, ***kwargs*)

Update a script

Parameters *id* : integer

The ID for the script.

schedule : dict, optional:

```
- scheduled : boolean
  If the object is scheduled
- scheduled_days : list
```

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

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.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
    ↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
    ↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
    ↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

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.

name : string, optional

The name of the script.

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

parent_id : integer, optional

The ID of the parent job that will trigger this script

sql : string, optional

The raw SQL query for the script.

notifications : dict, optional:

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

Returns schedule : dict:

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

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

sql : string

The raw SQL query for the script.

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳ field.
- required : boolean
  Whether this param is required.
- description : string
  A short sentence or fragment describing this parameter to the
↳ end user.
- name : string
  The variable's name as used within your code.
- value : string
  The value you would like to set this param to. Setting this
↳ value makes
  this parameter a fixed param.
- type : string
  The type of parameter. Valid options: string, integer, float,
↳ bool,
  file, database, credential_aws, credential_redshift, or
  credential_custom
- label : string
  The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

```
- started_at : string/time
  The time that the run started.
- id : integer
```

```

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

```

name : string

The name of the script.

template_script_id : integer

The ID of the template script, if any.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

updated_at : string/time

The time this script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

type : string

The type of script.

next_run_at : string/time

The time of the next scheduled run.

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.

```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```

A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.

```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

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

patch_containers (*id*, ***kwargs*)

Update a container

Parameters **id** : integer

The ID for the script.

schedule : dict, optional:

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

```
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

docker_image_tag : string, optional

The tag of the docker image to pull from DockerHub (default: latest).

repo_ref : string, optional

The tag or branch of the github repo to clone into the container.

name : string, optional

The name of the container.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments.
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

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.

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.
- 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.

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

docker_command : string, optional

The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]

time_zone : string, optional

The time zone of this script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

remote_host_credential_id : integer, optional

The id of the database credentials to pass into the environment of the container.

notifications : dict, optional:

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


docker_image_name : string, optional

The name of the docker image to pull from DockerHub.

Returns **schedule** : dict:

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

repo_ref : string

The tag or branch of the github repo to clone into the container.

hidden : boolean

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

author : dict:

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

repo_http_uri : string

The location of a github repo to clone into the container, e.g. github.com/my-user/my-repo.git.

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- required : boolean
    Whether this param is required.
```

```
- description : string
    A short sentence or fragment describing this parameter to the
    ↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
    ↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
    ↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the container.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template script.

type : string

The type of the script (e.g Container)

time_zone : string

The time zone of this script.

docker_image_name : string

The name of the docker image to pull from DockerHub.

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.
```

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

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.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

remote_host_credential_id : integer

The id of the database credentials to pass into the environment of the container.

id : integer

The ID for the script.

notifications : dict:

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

docker_command : string

The command to run on the container. Will be run via sh as: [`sh`”, `-c`”, `dockerCommand`]

patch_containers_runs (*id*, *run_id*, ***kwargs*)

Update a run

Parameters `id` : integer

The ID for the script.

run_id : integer

The ID of the script run.

state : string, optional

The state of the script.

bocce_started_at : string/date-time, optional

The time when a bocce worker began executing the script.

bocce_accepted_at : string/date-time, optional

The time when a bocce worker began processing the script.

Returns None

Response code 204: success

patch_custom (*id*, ***kwargs*)

Update some attributes of this CustomScript

Parameters `id` : integer

The ID for the script.

schedule : dict, optional:

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

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

name : string, optional

The name of the script.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

parent_id : integer, optional

The ID of the parent job that will trigger this script

time_zone : string, optional

The time zone of this script.

credential_id : integer, optional

The credential that this script will use.

notifications : dict, optional:

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

target_project_id : integer, optional

Target project to which script outputs will be added.

Returns **schedule** : dict:

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

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- required : boolean
  Whether this param is required.
- description : string
  A short sentence or fragment describing this parameter to the
↳end user.
- name : string
  The variable's name as used within your code.
- value : string
  The value you would like to set this param to. Setting this
↳value makes
  this parameter a fixed param.
- type : string
  The type of parameter. Valid options: string, integer, float,
↳bool,
  file, database, credential_aws, credential_redshift, or
  credential_custom
- label : string
  The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

last_run : dict:

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

```
    The time that the run was queued.
- state : string
- finished_at : string/time
    The time that the run completed.
```

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template script.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g Custom)

time_zone : string

The time zone of this script.

target_project_id : integer

Target project to which script outputs will be added.

code_preview : string

The code that this script will run with arguments inserted.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

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

patch_javascript (*id*, ****kwargs**)

Update some attributes of this JavaScript Script

Parameters *id* : integer

The ID for the script.

schedule : dict, optional:

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

name : string, optional

The name of the script.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

time_zone : string, optional

The time zone of this script.

credential_id : integer, optional

The credential that this script will use.

target_project_id : integer, optional

Target project to which script outputs will be added.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments.
↳ field.
- required : boolean
  Whether this param is required.
- description : string
  A short sentence or fragment describing this parameter to the
↳ end user.
- name : string
  The variable's name as used within your code.
- value : string
  The value you would like to set this param to. Setting this
↳ value makes
  this parameter a fixed param.
- type : string
  The type of parameter. Valid options: string, integer, float,
↳ bool,
  file, database, credential_aws, credential_redshift, or
  credential_custom
- label : string
  The label to present to users when asking them for the value.
- 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.
```

source : string, optional

The body/text 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.

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

```
- success_email_body : string
  Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
  If failure email notifications are on
- urls : list
  URLs to receive a POST request at job completion
- stall_warning_minutes : integer
  Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
  Addresses to notify by e-mail when the job fails.
```

```

- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_subject : string
    Custom subject line for success e-mail.

```

Returns `schedule` : dict:

```

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

```

hidden : boolean

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

author : dict:

```

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

```

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

params : list:

```

A definition of the parameters this script accepts in the arguments_
↳field.
- required : boolean

```

```
Whether this param is required.
- description : string
  A short sentence or fragment describing this parameter to the
  ↪end user.
- name : string
  The variable's name as used within your code.
- value : string
  The value you would like to set this param to. Setting this
  ↪value makes
  this parameter a fixed param.
- type : string
  The type of parameter. Valid options: string, integer, float,
  ↪bool,
  file, database, credential_aws, credential_redshift, or
  credential_custom
- label : string
  The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

```
- name : string
    This user's name.
```

id : integer

The ID for the script.

notifications : dict:

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

source : string

The body/text of the script.

patch_python3 (*id*, ***kwargs*)

Update some attributes of this Python Script

Parameters **id** : integer

The ID for the script.

schedule : dict, optional:

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

name : string, optional

The name of the script.

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

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.
```

params : list, optional:

```
A definition of the parameters this script accepts in the arguments.
    ↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
    ↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
    ↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
    ↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

source : string, optional

The body/text 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.

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

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

Returns **schedule** : dict:

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

hidden : boolean

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

author : dict:

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



```
- name : string
    This user's name.
```

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments,
↳field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↳value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↳bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

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

source : string

The body/text of the script.

patch_r (*id*, ***kwargs*)

Update some attributes of this R Script

Parameters *id* : integer

The ID for the script.

schedule : dict, optional:

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

name : string, optional

The name of the script.

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

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.
```

params : list, optional:

```

A definition of the parameters this script accepts in the arguments_
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,_
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.

```

source : string, optional

The body/text 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.

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

```

- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on
- urls : list
    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list

```

```
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
```

Returns `schedule` : dict:

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

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
```

```

    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.

```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

```

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

```

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

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

source : string

The body/text of the script.

patch_sql (*id*, ****kwargs**)

Update some attributes of this SQL script

Parameters **id** : integer

The ID for the script.

schedule : dict, optional:

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

name : string, optional

The name of the script.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

time_zone : string, optional

The time zone of this script.

sql : string, optional

The raw SQL query for the script.

credential_id : integer, optional

The credential that this script will use.

target_project_id : integer, optional

Target project to which script outputs will be added.

csv_settings : dict, optional:

```
- force_multifile : boolean
    Whether or not the csv should be split into multiple files.↵
↵Default:
    false
- column_delimiter : string
    Which delimiter to use, one of "comma", "tab", or "pipe".↵
↵Default:
    comma
- 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
- filename_prefix : string
    A user specified filename prefix for the output file to have.↵
↵Default:
    null
- unquoted : boolean
    Whether or not to quote fields. Default: false
```

params : list, optional:

```
A definition of the parameters this script accepts in the arguments.↵
↵field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the↵
↵end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this↵
↵value makes
```

```

    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.

```

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.

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

```

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

```

Returns **schedule** : dict:

```

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

```

```
Hours of the day it is scheduled on
- scheduled_minutes : list
  Minutes of the day it is scheduled on
```

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

sql : string

The raw SQL query for the script.

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

csv_settings : dict:

```
- force_multifile : boolean
  Whether or not the csv should be split into multiple files.
↪Default:
  false
- column_delimiter : string
  Which delimiter to use, one of "comma", "tab", or "pipe".
↪Default:
  comma
- 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
- filename_prefix : string
  A user specified filename prefix for the output file to have.
↪Default:
  null
```

```
- unquoted : boolean
    Whether or not to quote fields. Default: false
```

params : list:

A definition of the parameters this script accepts **in** the arguments.

```
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

```
- started_at : string/time
    The time that the run started.
- id : integer
- error : string
    The error message for this run, if present.
```

```
- created_at : string/time
    The time that the run was queued.
- state : string
- finished_at : string/time
    The time that the run completed.
```

name : string

The name of the script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

code_preview : string

The code that this script will run with arguments inserted.

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

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

post (*name*, *remote_host_id*, *sql*, *credential_id*, ***kwargs*)

Create a script

Parameters **name** : string

The name of the script.

remote_host_id : integer

The database ID.

sql : string

The raw SQL query for the script.

credential_id : integer

The credential ID.

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.  
- required : boolean  
  Whether this param is required.  
- description : string  
  A short sentence or fragment describing this parameter to the  
↪end user.  
- name : string  
  The variable's name as used within your code.  
- value : string  
  The value you would like to set this param to. Setting this  
↪value makes  
  this parameter a fixed param.  
- type : string  
  The type of parameter. Valid options: string, integer, float,  
↪bool,  
  file, database, credential_aws, credential_redshift, or  
  credential_custom  
- label : string  
  The label to present to users when asking them for the value.  
- 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.
```

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.

hidden : boolean, 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

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

notifications : dict, optional:

```
- success_email_body : string  
  Custom body text for success e-mail, written in Markdown.  
- failure_on : boolean  
  If failure email notifications are on  
- urls : list  
  URLs to receive a POST request at job completion  
- stall_warning_minutes : integer  
  Stall warning emails will be sent after this amount of minutes.  
- failure_email_addresses : list  
  Addresses to notify by e-mail when the job fails.  
- success_on : boolean  
  If success email notifications are on
```



```

- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.

```

Returns `schedule` : dict:

```

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

```

hidden : boolean

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

author : dict:

```

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

```

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

```

A definition of the parameters this script accepts in the arguments_
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- name : string
    The variable's name as used within your code.

```

```
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,_
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the script.

template_script_id : integer

The ID of the template script, if any.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

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

post_cancel (*id*)

Cancel a run

Parameters *id* : integer

The ID of the job.

Returns *state* : string

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

id : integer

The ID of the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

post_containers (*required_resources*, *docker_command*, *docker_image_name*, ***kwargs*)

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.

schedule : dict, optional:

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

docker_image_tag : string, optional

The tag of the docker image to pull from DockerHub (default: latest).

repo_ref : string, optional

The tag or branch of the github repo to clone into the container.

hidden : boolean, 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

name : string, optional

The name of the container.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
    ↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
    ↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
    ↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

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.

target_project_id : integer, optional

Target project to which script outputs will be added.

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.

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

time_zone : string, optional

The time zone of this script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

remote_host_credential_id : integer, optional

The id of the database credentials to pass into the environment of the container.

notifications : dict, optional:

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

Returns `schedule` : dict:

```

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

```

repo_ref : string

The tag or branch of the github repo to clone into the container.

hidden : boolean

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

author : dict:

```

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

```

repo_http_uri : string

The location of a github repo to clone into the container, e.g. github.com/my-user/my-repo.git.

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

```

A definition of the parameters this script accepts in the arguments_
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- name : string
    The variable's name as used within your code.

```

```
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,_
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the container.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template script.

type : string

The type of the script (e.g Container)

time_zone : string

The time zone of this script.

docker_image_name : string

The name of the docker image to pull from DockerHub.

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.
```

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

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.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

remote_host_credential_id : integer

The id of the database credentials to pass into the environment of the container.

id : integer

The ID for the script.

notifications : dict:

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

docker_command : string

The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]

post_containers_runs (*id*)

Start a run

Parameters *id* : integer

The ID of the container.

Returns *started_at* : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

finished_at : string/time

The time the last run completed.

container_id : integer

The ID of the container.

post_containers_runs_heartbeats (*id*, *run_id*)

Indicate that the given run is being handled

Parameters **id** : integer

The ID of the container.

run_id : integer

The ID of the run.

Returns None

Response code 204: success

post_containers_runs_logs (*id*, *run_id*, ****kwargs**)

Add log messages

Parameters **id** : integer

The ID of the script.

run_id : integer

The ID of the script run.

messages : list, optional:

```
- message : string
    The log message to store.
- created_at : string/date-time
- level : string
    The log level of this message [default: info]
```

message : string, optional

The log message to store.

level : string, optional

The log level of this message [default: info]

Returns None

Response code 204: success

post_containers_runs_outputs (*id, run_id, object_type, object_id*)

Add an output for a run

Parameters **id** : integer

The ID of the output.

run_id : integer

The ID of the run.

object_type : string

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

object_id : integer

The ID of the output object.

Returns **link** : string

The link to retrieve the output object.

name : string

The name of the output object.

object_type : string

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

object_id : integer

The ID of the output object.

post_custom (*from_template_id, **kwargs*)

Create a CustomScript

Parameters **from_template_id** : integer

The ID of the template script.

schedule : dict, optional:

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

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

hidden : boolean, 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

name : string, optional

The name of the script.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

parent_id : integer, optional

The ID of the parent job that will trigger this script

time_zone : string, optional

The time zone of this script.

credential_id : integer, optional

The credential that this script will use.

notifications : dict, optional:

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

target_project_id : integer, optional

Target project to which script outputs will be added.

Returns **schedule** : dict:

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

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↳value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↳bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

last_run : dict:

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

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template script.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g Custom)

time_zone : string

The time zone of this script.

target_project_id : integer

Target project to which script outputs will be added.

code_preview : string

The code that this script will run with arguments inserted.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

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

post_custom_runs (*id*)

Start a run

Parameters **id** : integer

The ID of the custom.

Returns **started_at** : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

custom_id : integer

The ID of the custom.

finished_at : string/time

The time the last run completed.

post_custom_runs_outputs (*id, run_id, object_type, object_id*)

Add an output for a run

Parameters id : integer

The ID of the output.

run_id : integer

The ID of the run.

object_type : string

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

object_id : integer

The ID of the output object.

Returns link : string

The link to retrieve the output object.

name : string

The name of the output object.

object_type : string

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

object_id : integer

The ID of the output object.

post_javascript (*name, remote_host_id, credential_id, source, **kwargs*)

Create a JavaScript Script

Parameters name : string

The name 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.

source : string

The body/text of the script.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_runs_per_hour : integer
```

```
    Alternative to scheduled minutes, number of times to run per_
↪hour
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

time_zone : string, optional

The time zone of this script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

target_project_id : integer, optional

Target project to which script outputs will be added.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,_
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

hidden : boolean, 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

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

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

Returns **schedule** : dict:

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

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

params : list:

A definition of the parameters this script accepts **in** the arguments.

- ↪ **field**.
- **required** : boolean
Whether this param **is** required.
- **description** : string
A short sentence **or** fragment describing this parameter to the ↪
↪ end user.
- **name** : string
The variable's name as used within your code.
- **value** : string
The value you would like to **set** this param to. Setting this ↪
↪ value makes
this parameter a fixed param.
- **type** : string
The **type** of parameter. Valid options: string, integer, float, ↪
↪ bool,
file, database, credential_aws, credential_redshift, **or**
credential_custom
- **label** : string
The label to present to users when asking them **for** the value.
- **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**.

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

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

source : string

The body/text of the script.

post_javascript_runs (*id*)

Start a run

Parameters *id* : integer

The ID of the javascript.

Returns *started_at* : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

javascript_id : integer

The ID of the javascript.

finished_at : string/time

The time the last run completed.

post_javascript_runs_outputs (*id, run_id, object_type, object_id*)

Add an output for a run

Parameters **id** : integer

The ID of the output.

run_id : integer

The ID of the run.

object_type : string

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

object_id : integer

The ID of the output object.

Returns **link** : string

The link to retrieve the output object.

name : string

The name of the output object.

object_type : string

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

object_id : integer

The ID of the output object.

post_python3 (*name, source, **kwargs*)

Create a Python Script

Parameters **name** : string

The name of the script.

source : string

The body/text of the script.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
```

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per
    ↪hour
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

time_zone : string, optional

The time zone of this script.

parent_id : integer, optional

The ID of the parent job that will trigger 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.
```

params : list, optional:

```
A definition of the parameters this script accepts in the arguments.
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
    ↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
    ↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
    ↪bool,
```



```

    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.

```

hidden : boolean, 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

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

```

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

```

Returns **schedule** : dict:

```

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

```

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments,
↳field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↳value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↳bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
```

```

- failure_on : boolean
    If failure email notifications are on
- urls : list
    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.

```

source : string

The body/text of the script.

post_python3_runs (*id*)

Start a run

Parameters *id* : integer

The ID of the python.

Returns *started_at* : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

finished_at : string/time

The time the last run completed.

python_id : integer

The ID of the python.

post_python3_runs_outputs (*id*, *run_id*, *object_type*, *object_id*)

Add an output for a run

Parameters *id* : integer

The ID of the output.

run_id : integer

The ID of the run.

object_type : string

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

object_id : integer

The ID of the output object.

Returns link : string

The link to retrieve the output object.

name : string

The name of the output object.

object_type : string

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

object_id : integer

The ID of the output object.

post_r (*name*, *source*, ***kwargs*)

Create an R Script

Parameters name : string

The name of the script.

source : string

The body/text of the script.

schedule : dict, optional:

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

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

time_zone : string, optional

The time zone of this script.

parent_id : integer, optional

The ID of the parent job that will trigger 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.

```

params : list, optional:

```

A definition of the parameters this script accepts in the arguments
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
    ↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
    ↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
    ↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.

```

hidden : boolean, 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

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

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

Returns **schedule** : dict:

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

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

A definition of the parameters this script accepts **in** the arguments.

- ↪field.
- required : boolean
 - Whether this param **is** required.
- description : string
 - A short sentence **or** fragment describing this parameter to the
 - ↪end user.
- name : string
 - The variable's name as used within your code.
- value : string
 - The value you would like to **set** this param to. Setting this
 - ↪value makes
 - this parameter a fixed param.
- type : string
 - The **type** of parameter. Valid options: string, integer, float,
 - ↪bool,
 - file, database, credential_aws, credential_redshift, **or**
 - credential_custom
- label : string
 - The label to present to users when asking them **for** the value.
- 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.

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

- started_at : string/time
 - The time that the run started.
- id : integer
- error : string
 - The error message **for** this run, **if** present.
- created_at : string/time

```
The time that the run was queued.
- state : string
- finished_at : string/time
  The time that the run completed.
```

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

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

source : string

The body/text of the script.

post_r_runs (*id*)

Start a run

Parameters **id** : integer

The ID of the r.

Returns **started_at** : string/time

The time the last run started at.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

state : string

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

r_id : integer

The ID of the r.

finished_at : string/time

The time the last run completed.

post_r_runs_outputs (*id, run_id, object_type, object_id*)

Add an output for a run

Parameters **id** : integer

The ID of the output.

run_id : integer

The ID of the run.

object_type : string

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

object_id : integer

The ID of the output object.

Returns **link** : string

The link to retrieve the output object.

name : string

The name of the output object.

object_type : string

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

object_id : integer

The ID of 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, remote_host_id, sql, credential_id, **kwargs*)

Create a SQL script

Parameters **name** : string

The name of the script.

remote_host_id : integer

The remote host ID that this script will connect to.

sql : string

The raw SQL query for the script.

credential_id : integer

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_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

time_zone : string, optional

The time zone of this script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

target_project_id : integer, optional

Target project to which script outputs will be added.

csv_settings : dict, optional:

```
- force_multifile : boolean
    Whether or not the csv should be split into multiple files.
↪Default:
    false
- column_delimiter : string
    Which delimiter to use, one of "comma", "tab", or "pipe".
↪Default:
    comma
- 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
- filename_prefix : string
  A user specified filename prefix for the output file to have.↵
↵Default:
  null
- unquoted : boolean
  Whether or not to quote fields. Default: false
```

params : list, optional:

```
A definition of the parameters this script accepts in the arguments.↵
↵field.
- required : boolean
  Whether this param is required.
- description : string
  A short sentence or fragment describing this parameter to the↵
↵end user.
- name : string
  The variable's name as used within your code.
- value : string
  The value you would like to set this param to. Setting this↵
↵value makes
  this parameter a fixed param.
- type : string
  The type of parameter. Valid options: string, integer, float,↵
↵bool,
  file, database, credential_aws, credential_redshift, or
  credential_custom
- label : string
  The label to present to users when asking them for the value.
- 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.
```

hidden : boolean, 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

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has
defined params.

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

```

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

```

Returns `schedule` : dict:

```

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

```

hidden : boolean

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

author : dict:

```

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

```

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

sql : string

The raw SQL query for the script.

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

csv_settings : dict:

```
- force_multifile : boolean
    Whether or not the csv should be split into multiple files.
↪Default:
    false
- column_delimiter : string
    Which delimiter to use, one of "comma", "tab", or "pipe".
↪Default:
    comma
- 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
- filename_prefix : string
    A user specified filename prefix for the output file to have.
↪Default:
    null
- unquoted : boolean
    Whether or not to quote fields. Default: false
```

params : list:

```
A definition of the parameters this script accepts in the arguments.
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.

```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

```

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

```

name : string

The name of the script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

code_preview : string

The code that this script will run with arguments inserted.

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on
- urls : list
    URLs to receive a POST request at job completion
```

```

- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.

```

post_sql_runs (*id*)

Start a run

Parameters *id* : integer

The ID of the sql.

Returns *started_at* : string/time

The time the last run started.

id : integer

The ID of this run.

error : string

The error message for this run, if present.

is_cancel_requested : boolean

True if run cancel requested, else false.

sql_id : integer

The ID of this sql.

state : string

The state of this run.

output : list:

```

A list of the outputs of this script.
- path : string
    The temporary link to download this output file, valid for 36_
↪hours.
- file_id : integer
    The unique ID of the output file.
- output_name : string
    The name of the output file.

```

finished_at : string/time

The time that this run finished.

put_containers (*id*, *required_resources*, *docker_command*, *docker_image_name*, ***kwargs*)

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.

schedule : dict, optional:

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

docker_image_tag : string, optional

The tag of the docker image to pull from DockerHub (default: latest).

repo_ref : string, optional

The tag or branch of the github repo to clone into the container.

name : string, optional

The name of the container.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments.
    ↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
    ↪end user.
- name : string
```

```

    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.

```

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

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.

target_project_id : integer, optional

Target project to which script outputs will be added.

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.

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

time_zone : string, optional

The time zone of this script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

remote_host_credential_id : integer, optional

The id of the database credentials to pass into the environment of the container.

notifications : dict, optional:

```

- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on

```

```
- urls : list
    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
```

Returns `schedule` : dict:

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

repo_ref : string

The tag or branch of the github repo to clone into the container.

hidden : boolean

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

author : dict:

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

repo_http_uri : string

The location of a github repo to clone into the container, e.g. `github.com/my-user/my-repo.git`.

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- required : boolean
  Whether this param is required.
- description : string
  A short sentence or fragment describing this parameter to the
↳end user.
- name : string
  The variable's name as used within your code.
- value : string
  The value you would like to set this param to. Setting this
↳value makes
  this parameter a fixed param.
- type : string
  The type of parameter. Valid options: string, integer, float,
↳bool,
  file, database, credential_aws, credential_redshift, or
  credential_custom
- label : string
  The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

```
- started_at : string/time
  The time that the run started.
- id : integer
- error : string
```

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

name : string

The name of the container.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template script.

type : string

The type of the script (e.g Container)

time_zone : string

The time zone of this script.

docker_image_name : string

The name of the docker image to pull from DockerHub.

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.
```

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

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.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

remote_host_credential_id : integer

The id of the database credentials to pass into the environment of the container.

id : integer

The ID for the script.

notifications : dict:

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

```
- success_email_subject : string
    Custom subject line for success e-mail.
```

docker_command : string

The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]

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 *schedule* : dict:

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

repo_ref : string

The tag or branch of the github repo to clone into the container.

hidden : boolean

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

author : dict:

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

repo_http_uri : string

The location of a github repo to clone into the container, e.g. github.com/my-user/my-repo.git.

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳ field.
- required : boolean
  Whether this param is required.
- description : string
  A short sentence or fragment describing this parameter to the
↳ end user.
- name : string
  The variable's name as used within your code.
- value : string
  The value you would like to set this param to. Setting this
↳ value makes
  this parameter a fixed param.
- type : string
  The type of parameter. Valid options: string, integer, float,
↳ bool,
  file, database, credential_aws, credential_redshift, or
  credential_custom
- label : string
  The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

```
- started_at : string/time
  The time that the run started.
- id : integer
```

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

name : string

The name of the container.

docker_image_tag : string

The tag of the docker image to pull from DockerHub (default: latest).

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template script.

type : string

The type of the script (e.g Container)

time_zone : string

The time zone of this script.

docker_image_name : string

The name of the docker image to pull from DockerHub.

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.
```

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

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.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

remote_host_credential_id : integer

The id of the database credentials to pass into the environment of the container.

id : integer

The ID for the script.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on
- urls : list
    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
```

```
Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_subject : string
  Custom subject line for success e-mail.
```

docker_command : string

The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]

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*, *permission_level*, *group_ids*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

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

group_ids : list

An array of one or more group IDs

Returns **total_group_shares** : integer

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

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
```

readers : 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.

put_containers_shares_users (*id, user_ids, permission_level*)

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”

Returns **total_group_shares** : integer

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

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

```

readers : 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.

put_custom(*id*, ***kwargs*)

Replace all attributes of this CustomScript

Parameters *id* : integer

The ID for the script.

schedule : dict, optional:

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

arguments : dict, optional

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

name : string, optional

The name of the script.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

parent_id : integer, optional

The ID of the parent job that will trigger this script

time_zone : string, optional

The time zone of this script.

credential_id : integer, optional

The credential that this script will use.

notifications : dict, optional:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on
- urls : list
    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
```



```

    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.

```

target_project_id : integer, optional

Target project to which script outputs will be added.

Returns **schedule** : dict:

```

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

```

hidden : boolean

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

author : dict:

```

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

```

state : string

The status of the script's last run.

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

params : list:

```

A definition of the parameters this script accepts in the arguments_
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.

```

```
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,_
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

last_run : dict:

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

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template script.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g Custom)

time_zone : string

The time zone of this script.

target_project_id : integer

Target project to which script outputs will be added.

code_preview : string

The code that this script will run with arguments inserted.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on
- urls : list
    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
```

```
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
```

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 *schedule* : dict:

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

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

params : list:

```

A definition of the parameters this script accepts in the arguments_
↳field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the_
↳end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this_
↳value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,_
↳bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.

```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

last_run : dict:

```

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

```

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template script.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g Custom)

time_zone : string

The time zone of this script.

target_project_id : integer

Target project to which script outputs will be added.

code_preview : string

The code that this script will run with arguments inserted.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on
- urls : list
```

```

    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
    successfully.
- success_email_subject : string
    Custom subject line for success e-mail.

```

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*, *permission_level*, *group_ids*)

Set the permissions groups has on this object

Parameters *id* : integer

ID of the resource to be shared

permission_level : string

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

group_ids : list

An array of one or more group IDs

Returns **total_group_shares** : integer

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

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
```

readers : 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.

put_custom_shares_users (*id*, *user_ids*, *permission_level*)

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”

Returns **total_group_shares** : integer

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

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
```

readers : 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.

put_javascript (*id, name, remote_host_id, credential_id, source, **kwargs*)

Replace all attributes of this JavaScript Script

Parameters **id** : integer

The ID for the script.

name : string

The name 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.

source : string

The body/text of the script.

schedule : dict, optional:

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

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

time_zone : string, optional

The time zone of this script.

target_project_id : integer, optional

Target project to which script outputs will be added.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
```

```
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,_
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

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.

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

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

Returns `schedule` : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_days : list
```

```

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

```

hidden : boolean

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

author : dict:

```

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

```

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

params : list:

```

A definition of the parameters this script accepts in the arguments_
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,_
↪bool,

```

```
file, database, credential_aws, credential_redshift, or
credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on
```

```
- urls : list
    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
```

source : string

The body/text of the script.

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 *schedule* : dict:

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

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- required : boolean
  Whether this param is required.
- description : string
  A short sentence or fragment describing this parameter to the
↳end user.
- name : string
  The variable's name as used within your code.
- value : string
  The value you would like to set this param to. Setting this
↳value makes
  this parameter a fixed param.
- type : string
  The type of parameter. Valid options: string, integer, float,
↳bool,
  file, database, credential_aws, credential_redshift, or
  credential_custom
- label : string
  The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

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

source : string

The body/text of the script.

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, permission_level, group_ids*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

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

group_ids : list

An array of one or more group IDs

Returns **total_group_shares** : integer

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

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
```

readers : 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.

put_javascript_shares_users (*id, user_ids, permission_level*)

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”

Returns `total_group_shares` : integer

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

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
```

readers : 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.

put_python3 (*id*, *name*, *source*, ***kwargs*)

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.

schedule : dict, optional:

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

```
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

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.
```

params : list, optional:

```
A definition of the parameters this script accepts in the arguments.
    ↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
    ↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
    ↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
    ↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.

```

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.

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

```

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

```

Returns schedule : dict:

```

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

```

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
  The ID for the project.
- name : string
  The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

```
- success_email_body : string
  Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
  If failure email notifications are on
- urls : list
  URLs to receive a POST request at job completion
- stall_warning_minutes : integer
  Stall warning emails will be sent after this amount of minutes.
```



```

- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳ successfully.
- success_email_subject : string
    Custom subject line for success e-mail.

```

source : string

The body/text of the script.

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 **schedule** : dict:

```

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

```

hidden : boolean

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

author : dict:

```

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

```

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

A definition of the parameters this script accepts **in** the arguments.
↪field.
- required : boolean
 Whether this param **is** required.
- description : string
 A short sentence **or** fragment describing this parameter to the
↪end user.
- name : string
 The variable's name as used within your code.
- value : string
 The value you would like to **set** this param to. Setting this
↪value makes
 this parameter a fixed param.
- type : string
 The **type** of parameter. Valid options: string, integer, float,
↪bool,
 file, database, credential_aws, credential_redshift, **or**
 credential_custom
- label : string
 The label to present to users when asking them **for** the value.
- 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**.

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

- started_at : string/time
 The time that the run started.
- id : integer
- error : string

```

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

```

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.

```

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.

```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.  
- id : integer  
    The ID for the project.  
- name : string  
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

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

source : string

The body/text of the script.

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, permission_level, group_ids*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

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

group_ids : list

An array of one or more group IDs

Returns **total_group_shares** : integer

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

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
```

readers : 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.

put_python3_shares_users (*id, user_ids, permission_level*)

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”

Returns **total_group_shares** : integer

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

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
```

readers : 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.

put_r (*id, name, source, **kwargs*)

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.

schedule : dict, optional:

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

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

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.
```

params : list, optional:

```
A definition of the parameters this script accepts in the arguments._
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this_
↪value makes
```

```
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

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.

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

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

Returns **schedule** : dict:

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



```

    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on

```

hidden : boolean

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

author : dict:

```

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

```

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

```

A definition of the parameters this script accepts in the arguments.
↳field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↳value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↳bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

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.

```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```

A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.

```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

```

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

```

id : integer

The ID for the script.

notifications : dict:

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

source : string

The body/text of the script.

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 **schedule** : dict:

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

hidden : boolean

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

author : dict:

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

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

finished_at : string/time

The time that the script's last run finished.

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↳value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↳bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the script.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

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.
```

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on
- urls : list
    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
    ↪ successfully.
```

```
- success_email_subject : string
    Custom subject line for success e-mail.
```

source : string

The body/text of the script.

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*, *permission_level*, *group_ids*)

Set the permissions groups has on this object

Parameters *id* : integer

ID of the resource to be shared

permission_level : string

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

group_ids : list

An array of one or more group IDs

Returns **total_group_shares** : integer

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

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
```

readers : 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.

put_r_shares_users (*id*, *user_ids*, *permission_level*)

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”

Returns **total_group_shares** : integer

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

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

```

readers : 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.

put_sql (*id, name, remote_host_id, sql, credential_id, **kwargs*)

Replace all attributes of this SQL script

Parameters **id** : integer

The ID for the script.

name : string

The name of the script.

remote_host_id : integer

The remote host ID that this script will connect to.

sql : string

The raw SQL query for the script.

credential_id : integer

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_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per_
↪hour
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

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:

```
- force_multifile : boolean
    Whether or not the csv should be split into multiple files.↪
↪Default:
    false
- column_delimiter : string
    Which delimiter to use, one of "comma", "tab", or "pipe".↪
↪Default:
    comma
- 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
- filename_prefix : string
    A user specified filename prefix for the output file to have.
↳Default:
    null
- unquoted : boolean
    Whether or not to quote fields. Default: false

```

params : list, optional:

A definition of the parameters this script accepts in the arguments.
↳field.

```

- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↳value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↳bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.

```

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.

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

```

- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on

```

```
- urls : list
    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
```

Returns `schedule` : dict:

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

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

sql : string

The raw SQL query for the script.

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

csv_settings : dict:

```
- force_multifile : boolean
    Whether or not the csv should be split into multiple files.
↪Default:
    false
- column_delimiter : string
    Which delimiter to use, one of "comma", "tab", or "pipe".
↪Default:
    comma
- 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
- filename_prefix : string
    A user specified filename prefix for the output file to have.
↪Default:
    null
- unquoted : boolean
    Whether or not to quote fields. Default: false
```

params : list:

```
A definition of the parameters this script accepts in the arguments.
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

code_preview : string

The code that this script will run with arguments inserted.

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on
- urls : list
    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
```

```
Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_subject : string
  Custom subject line for success e-mail.
```

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 *schedule* : dict:

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

hidden : boolean

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

author : dict:

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

state : string

The status of the script's last run.

parent_id : integer

The ID of the parent job that will trigger this script

sql : string

The raw SQL query for the script.

credential_id : integer

The credential that this script will use.

finished_at : string/time

The time that the script's last run finished.

csv_settings : dict:

```
- force_multifile : boolean
    Whether or not the csv should be split into multiple files.
↪Default:
    false
- column_delimiter : string
    Which delimiter to use, one of "comma", "tab", or "pipe".
↪Default:
    comma
- 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
- filename_prefix : string
    A user specified filename prefix for the output file to have.
↪Default:
    null
- unquoted : boolean
    Whether or not to quote fields. Default: false
```

params : list:

```
A definition of the parameters this script accepts in the arguments.
↪field.
- required : boolean
    Whether this param is required.
- description : string
    A short sentence or fragment describing this parameter to the
↪end user.
- name : string
    The variable's name as used within your code.
- value : string
    The value you would like to set this param to. Setting this
↪value makes
    this parameter a fixed param.
- type : string
    The type of parameter. Valid options: string, integer, float,
↪bool,
    file, database, credential_aws, credential_redshift, or
    credential_custom
- label : string
    The label to present to users when asking them for the value.
- 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.
```

template_script_name : string

The name of the template script.

archived : string

The archival status of the requested object(s).

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.

target_project_id : integer

Target project to which script outputs will be added.

last_run : dict:

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

name : string

The name of the script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

updated_at : string/time

The time the script was last updated.

from_template_id : integer

The ID of the template this script uses, if any.

remote_host_id : integer

The remote host ID that this script will connect to.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

next_run_at : string/time

The time of the next scheduled run.

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.
```

code_preview : string

The code that this script will run with arguments inserted.

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

is_template : boolean

Whether others scripts use this one as a template.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

created_at : string/time

The time this script was created.

time_zone : string

The time zone of this script.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

running_as : dict:

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

id : integer

The ID for the script.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- failure_on : boolean
    If failure email notifications are on
- urls : list
    URLs to receive a POST request at job completion
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- success_email_addresses : list
```

```
Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_subject : string
  Custom subject line for success e-mail.
```

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*, *permission_level*, *group_ids*)

Set the permissions groups has on this object

Parameters *id* : integer

ID of the resource to be shared

permission_level : string

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

group_ids : list

An array of one or more group IDs

Returns *total_group_shares* : integer

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

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
```

readers : 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.

put_sql_shares_users (*id*, *user_ids*, *permission_level*)

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”

Returns **total_group_shares** : integer

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

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
```

readers : 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.

Tables

class Tables (*session*, *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
<code>get_enhancements_geocodings(id, source_table_id)</code>	View the status of a geocoding table enhancement
<code>get_enhancements_prepared_matchings(id, ...)</code>	View a prepared matching enhancement
<code>get_enhancements_table_matchings(id, ...)</code>	View a table matching enhancement
<code>list(**kwargs)</code>	List tables
<code>list_columns(id, **kwargs)</code>	List columns in the specified table
<code>patch(id, **kwargs)</code>	Update a table
<code>post(database_id, data, schema, name)</code>	Import a file into a table
<code>post_enhancements_cass_ncoa(source_table_id, ...)</code>	Standardize addresses in a table
<code>post_enhancements_geocodings(source_table_id, ...)</code>	Geocode a table
<code>post_enhancements_prepared_matchings(...)</code>	Match person records against a dynamo table prepared by Civis
<code>post_enhancements_table_matchings(...)</code>	Match person records against an arbitrary Redshift table
<code>post_refresh(id)</code>	Request a refresh for column and table statistics

get (*id*)

Show basic table info

Parameters *id* : integer

Returns *database_id* : integer

The ID of the database.

description : string

The description of the table, as specified by the table owner

id : integer

The ID of the table.

row_count : integer

The number of rows in the table.

last_refresh : string/date-time

The time of the last statistics refresh.

enhancements : list:

- *type* : string
- *join_id* : integer
- *updated_at* : string/time
- *created_at* : string/time

size_mb : number/float

The size of the table in megabytes.

column_count : integer

The number of columns in the table.

columns : list:

```
- sql_type : string
    SQL type of the column.
- avg_value : number/float
    Average value of the column, where applicable.
- coverage_count : integer
    Number of non-null values in the column.
- description : string
    The description of the column, as specified by the table owner
- name : string
    Name of the column.
- order : integer
    Relative position of the column in the table.
- useable_as_primary_key : boolean
    Whether the column may be used as an primary key to identify
    ↪table
    rows.
- stddev : number/float
    Stddev of the column, where applicable.
- distinct_count : integer
    Number of distinct values in the column.
- max_value : string
    Largest value 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.
- value_distribution : dict
    An object mapping distinct values in the column to the number
    ↪of times
    they appear in the column
- min_value : string
    Smallest value in the column.
- sample_values : list
    A sample of values from the column.
- 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.
- useable_as_independent_variable : boolean
    Whether the column may be used as an independent variable to
    ↪train a
    model.
- null_count : integer
    Number of null values in the column.
- encoding : string
    The compression encoding for this columnSee: http://docs.aws.
    ↪amazon.com
```

```
/redshift/latest/dg/c_Compression_encodings.html
```

multipart_key : list

last_run : dict:

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

schema : string

The name of the schema containing the table.

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.

name : string

Name of the table.

ontology_mapping : dict

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

is_view : boolean

True if this table represents a view. False if it represents a regular table.

outgoing_table_matches : list:

```
- source_table_id : integer
    Source table
- job : dict::
    - match_options : dict::
        - max_matches : integer
        - threshold : string
    - runs : list::
        Information about the most recent runs of the job.
        - started_at : string/time
            The time that the run started.
        - id : integer
        - error : string
            The error message for this run, if present.
        - created_at : string/time
            The time that the run was queued.
        - state : string
        - finished_at : string/time
            The time that the run completed.
    - hidden : boolean
        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
    - id : integer
    - updated_at : string/date-time
    - type : string
    - state : string
      Whether the job is idle, queued, running, cancelled, or
↳ failed.
    - last_run : dict::
      - started_at : string/time
        The time that the run started.
      - id : integer
      - error : string
        The error message for this run, if present.
      - created_at : string/time
        The time that the run was queued.
      - state : string
      - finished_at : string/time
        The time that the run completed.
    - created_at : string/date-time
    - name : string
  - target_id : integer
    Target ID
  - target_type : string
    Target type
  - target : dict::
    - name : string

```

distkey : string

The column used as the Amazon Redshift distkey.

view_def : string

refresh_id : string

The ID of the most recent statistics refresh.

owner : string

The database username of the table's owner.

sortkeys : string

The column used as the Amazon Redshift sortkey.

joins : list:

```

- on : string
- left_table_id : integer
- left_join : boolean
- id : integer
- updated_at : string/time
- right_table_id : integer
- left_identifier : string
- created_at : string/time
- right_identifier : 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 **source_table_id** : integer

The ID of the table that was enhanced.

perform_ncoa : boolean

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

id : integer

The ID of the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

state : string

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

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.

ncoa_credential_id : integer

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

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 **enhanced_table_schema** : string

The schema name of the table created by the enhancement.

source_table_id : integer

The ID of the table that was enhanced.

id : integer

The ID of the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

state : string

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

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 *source_table_id* : integer

The ID of the table that was enhanced.

id : integer

The ID of the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

state : string

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

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.

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.

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 *source_table_id* : integer

The ID of the table that was enhanced.

id : integer

The ID of the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

state : string

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

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.

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.

list (***kwargs*)

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 `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.

database_id : integer

The ID of the database.

description : string

The description of the table, as specified by the table owner

name : string

Name of the table.

row_count : integer

The number of rows in the table.

last_refresh : string/date-time

The time of the last statistics refresh.

size_mb : number/float

The size of the table in megabytes.

last_run : dict:

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

is_view : boolean

True if this table represents a view. False if it represents a regular table.

column_count : integer

The number of columns in the table.

distkey : string

The column used as the Amazon Redshift distkey.

refresh_id : string

The ID of the most recent statistics refresh.

owner : string

The database username of the table's owner.

id : integer

The ID of the table.

sortkeys : string

The column used as the Amazon Redshift sortkey.

schema : string

The name of the schema containing the table.

list_columns (*id*, ***kwargs*)

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 **sql_type** : string

SQL type of the column.

avg_value : number/float

Average value of the column, where applicable.

coverage_count : integer

Number of non-null values in the column.

description : string

The description of the column, as specified by the table owner

name : string

Name of the column.

order : integer

Relative position of the column in the table.

useable_as_primary_key : boolean

Whether the column may be used as an primary key to identify table rows.

stddev : number/float

Stddev of the column, where applicable.

distinct_count : integer

Number of distinct values in the column.

max_value : string

Largest value 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.

value_distribution : dict

An object mapping distinct values in the column to the number of times they appear in the column

min_value : string

Smallest value in the column.

sample_values : list

A sample of values from the column.

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.

useable_as_independent_variable : boolean

Whether the column may be used as an independent variable to train a model.

null_count : integer

Number of null values in the column.

encoding : string

The compression encoding for this column. See: http://docs.aws.amazon.com/redshift/latest/dg/c_Compression_encodings.html

patch (*id*, ***kwargs*)

Update a table

Parameters **id** : integer

The ID of the table.

description : string, optional

The user-defined description of the table.

ontology_mapping : dict, optional

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

Returns **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.

database_id : integer

The ID of the database.

description : string

The description of the table, as specified by the table owner

ontology_mapping : dict

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

name : string

Name of the table.

row_count : integer

The number of rows in the table.

last_refresh : string/date-time

The time of the last statistics refresh.

size_mb : number/float

The size of the table in megabytes.

last_run : dict:

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

is_view : boolean

True if this table represents a view. False if it represents a regular table.

column_count : integer

The number of columns in the table.

distkey : string

The column used as the Amazon Redshift distkey.

refresh_id : string

The ID of the most recent statistics refresh.

owner : string

The database username of the table's owner.

id : integer

The ID of the table.

sortkeys : string

The column used as the Amazon Redshift sortkey.

schema : string

The name of the schema containing the table.

post (*database_id*, *data*, *schema*, *name*)

Import a file into a table

Parameters **database_id** : integer

The ID of the destination database.

data : string

The file to import, uploaded using HTTP multipart.

schema : string

The destination schema name.

name : string

The destination table name, without the schema prefix.

Returns **started_at** : string/date-time

The start time of the last run.

database_id : integer

The ID of the destination database.

schema : string

The destination schema name.

state : string

The state of the last run.

finished_at : string/date-time

The end time of the last run.

name : string

The destination table name, without the schema prefix.

post_enhancements_cass_ncoa (*source_table_id*, ***kwargs*)

Standardize addresses in a table

Parameters **source_table_id** : integer

The ID of the table to be enhanced.

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.

ncoa_credential_id : integer, optional

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

perform_ncoa : boolean, optional

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

Returns `source_table_id` : integer

The ID of the table that was enhanced.

perform_ncoa : boolean

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

id : integer

The ID of the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

state : string

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

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.

ncoa_credential_id : integer

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

post_enhancements_geocodings (*source_table_id*)

Geocode a table

Parameters `source_table_id` : integer

The ID of the table to be enhanced.

Returns `enhanced_table_schema` : string

The schema name of the table created by the enhancement.

source_table_id : integer

The ID of the table that was enhanced.

id : integer

The ID of the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

state : string

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

post_enhancements_prepared_matchings (*source_table_id*, *threshold*, *match_table_id*,
***kwargs*)

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 source_table_id : integer

The ID of the table that was enhanced.

id : integer

The ID of the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

state : string

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

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.

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.

post_enhancements_table_matchings (*source_table_id*, *threshold*, *match_table_id*,
***kwargs*)

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 `source_table_id` : integer

The ID of the table that was enhanced.

id : integer

The ID of the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

state : string

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

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.

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.

post_refresh (*id*)

Request a refresh for column and table statistics

Parameters `id` : integer

Returns `database_id` : integer

The ID of the database.

description : string

The description of the table, as specified by the table owner

id : integer

The ID of the table.

row_count : integer

The number of rows in the table.

last_refresh : string/date-time

The time of the last statistics refresh.

enhancements : list:

```
- type : string
- join_id : integer
- updated_at : string/time
- created_at : string/time
```

size_mb : number/float

The size of the table in megabytes.

column_count : integer

The number of columns in the table.

columns : list:

```
- sql_type : string
    SQL type of the column.
- avg_value : number/float
    Average value of the column, where applicable.
- coverage_count : integer
    Number of non-null values in the column.
- description : string
    The description of the column, as specified by the table owner
- name : string
    Name of the column.
- order : integer
    Relative position of the column in the table.
- useable_as_primary_key : boolean
    Whether the column may be used as an primary key to identify
↳table
    rows.
- stddev : number/float
    Stddev of the column, where applicable.
- distinct_count : integer
    Number of distinct values in the column.
- max_value : string
    Largest value 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.
- value_distribution : dict
    An object mapping distinct values in the column to the number
↳of times
    they appear in the column
- min_value : string
    Smallest value in the column.
- sample_values : list
    A sample of values from the column.
- 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.
- useable_as_independent_variable : boolean
    Whether the column may be used as an independent variable to
↳train a
    model.
- null_count : integer
    Number of null values in the column.
- encoding : string
```

```
The compression encoding for this columnSee: http://docs.aws.amazon.com/redshift/latest/dg/c\_Compression\_encodings.html
```

multipart_key : list

last_run : dict:

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

schema : string

The name of the schema containing the table.

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.

name : string

Name of the table.

ontology_mapping : dict

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

is_view : boolean

True if this table represents a view. False if it represents a regular table.

outgoing_table_matches : list:

```
- source_table_id : integer
    Source table
- job : dict::
    - match_options : dict::
        - max_matches : integer
        - threshold : string
    - runs : list::
        Information about the most recent runs of the job.
        - started_at : string/time
            The time that the run started.
        - id : integer
        - error : string
            The error message for this run, if present.
        - created_at : string/time
            The time that the run was queued.
        - state : string
        - finished_at : string/time
            The time that the run completed.
    - hidden : boolean
```

```

    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
    - id : integer
    - updated_at : string/date-time
    - type : string
    - state : string
        Whether the job is idle, queued, running, cancelled, or
↪failed.
    - last_run : dict::
        - started_at : string/time
            The time that the run started.
        - id : integer
        - error : string
            The error message for this run, if present.
        - created_at : string/time
            The time that the run was queued.
        - state : string
        - finished_at : string/time
            The time that the run completed.
    - created_at : string/date-time
    - name : string
- target_id : integer
    Target ID
- target_type : string
    Target type
- target : dict::
    - name : string

```

distkey : string

The column used as the Amazon Redshift distkey.

view_def : string

refresh_id : string

The ID of the most recent statistics refresh.

owner : string

The database username of the table's owner.

sortkeys : string

The column used as the Amazon Redshift sortkey.

joins : list:

```

- on : string
- left_table_id : integer
- left_join : boolean
- id : integer
- updated_at : string/time
- right_table_id : integer
- left_identifier : string
- created_at : string/time
- right_identifier : string

```

Users

class Users (*session*, *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(**kwargs)</code>	List users
<code>list_api_keys(id, **kwargs)</code>	Show API keys belonging to the specified user
<code>list_me()</code>	Show info about the logged-in user
<code>patch_me(**kwargs)</code>	Update info about the logged-in user
<code>post_api_keys(id, expires_in, name, **kwargs)</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 **expired** : boolean

True if the key has expired.

use_count : integer

The number of times the key has been used.

expires_at : string/date-time

The date and time when the key expired.

scopes : list

The scopes which the key is permissioned on.

last_used_at : string/date-time

The date and time when the key was last used.

active : boolean

True if the key has neither expired nor been revoked.

constraints : list:

```
Constraints on the abilities of the created key
- post_allowed : boolean
    Whether the constraint allows POST requests.
- head_allowed : boolean
    Whether the constraint allows HEAD requests.
- delete_allowed : boolean
    Whether the constraint allows DELETE requests.
- put_allowed : boolean
    Whether the constraint allows PUT requests.
```



```

- 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.
- patch_allowed : boolean
    Whether the constraint allows PATCH requests.

```

revoked_at : string/date-time

The date and time when the key was revoked.

name : string

The name of the API key.

created_at : string/date-time

The date and time when the key was created.

id : integer

The ID of the API key.

get (*id*)

Show info about a user

Parameters **id** : integer

The ID of this user.

Returns **city** : string

The city of this user.

title : string

The title of this user.

vpn_enabled : string

The availability of vpn for this user.

id : integer

The ID of this user.

phone : string

The phone number of this user.

time_zone : string

The time zone of this user.

department : string

The deartment of this user.

github_username : string

The GitHub username of this user.

prefers_sms_otp : string

The preference for phone authorization of this user

user : string

The username of this user.

initials : string

The initials of this user.

otp_required_for_login : string

The two factor authorization requirement for this user.

email : string

The email of this user.

state : string

The state 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**.

- **organization_id** : integer
The organization associated **with** this group.
- **id** : integer
The ID of this group.
- **name** : string
The name of this group.

name : string

The name 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 **expired** : boolean

True if the key has expired.

use_count : integer

The number of times the key has been used.

expires_at : string/date-time

The date and time when the key expired.

scopes : list

The scopes which the key is permissioned on.

last_used_at : string/date-time

The date and time when the key was last used.

active : boolean

True if the key has neither expired nor been revoked.

constraints : list:

Constraints on the abilities of the created key

- **post_allowed** : boolean
Whether the constraint allows POST requests.
- **head_allowed** : boolean
Whether the constraint allows HEAD requests.
- **delete_allowed** : boolean
Whether the constraint allows DELETE requests.
- **put_allowed** : boolean
Whether the constraint allows PUT requests.
- **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.
- **patch_allowed** : boolean
Whether the constraint allows PATCH requests.

revoked_at : string/date-time

The date and time when the key was revoked.

name : string

The name of the API key.

created_at : string/date-time

The date and time when the key was created.

id : integer

The ID of the API key.

list (**kwargs)

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.

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 **active** : string

The account status of this user.

id : integer

The ID of this user.

email : string

The email of this user.

user : string

The username of this user.

primary_group_id : integer

The ID of the primary group of this user.

current_sign_in_at : string/date-time

The date and time when the user's current session began.

groups : list:

An array of **all** the groups this user **is in**.

- **organization_id** : integer
The organization associated **with** this group.
- **id** : integer
The ID of this group.
- **name** : string
The name of this group.

created_at : string/date-time

The date and time when the user was created.

name : string

The name of this user.

list_api_keys (*id*, ***kwargs*)

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 expired : boolean

True if the key has expired.

use_count : integer

The number of times the key has been used.

constraint_count : integer

The number of constraints on the created key

expires_at : string/date-time

The date and time when the key expired.

scopes : list

The scopes which the key is permissioned on.

last_used_at : string/date-time

The date and time when the key was last used.

active : boolean

True if the key has neither expired nor been revoked.

revoked_at : string/date-time

The date and time when the key was revoked.

name : string

The name of the API key.

created_at : string/date-time

The date and time when the key was created.

id : integer

The ID of the API key.

list_me()

Show info about the logged-in user

Returns **username** : string

This user's username.

preferences : dict

This user's preferences.

organization_name : string

The name of the organization the user belongs to.

id : integer

The ID of this user.

feature_flags : dict

The feature flag settings for this user.

initials : string

This user's initials.

custom_branding : string

The branding of Platform for this user.

roles : list

The roles this user has, listed by slug.

email : string

This user's email address.

last_checked_announcements : string/date-time

The date and time at which the user last checked their announcements.

groups : list:

An array of **all** the groups this user **is in**.

- **organization_id** : integer
The organization associated **with** this group.
- **id** : integer
The ID of this group.
- **name** : string
The name of this group.

name : string

This user's name.

patch_me(kwargs)**

Update info about the logged-in user

Parameters **last_checked_announcements** : string/date-time, optional

The date and time at which the user last checked their announcements.

preferences : dict, optional:

```

- project_index_archived_filter : string
    Archived filter for the projects index page.
- export_index_author_filter : string
    Author filter for the exports index page.
- export_index_order_field : string
    Order field for the exports index page.
- model_index_thumbnail_view : string
    Thumbnail view for the models index page.
- model_index_order_dir : string
    Order direction for the models index page.
- result_index_archived_filter : string
    Archived filter for the results index page.
- project_detail_archived_filter : string
    Archived filter for the projects detail pages.
- project_detail_order_field : string
    Order field for projects detail pages.
- export_index_status_filter : string
    Status filter for the exports index page.
- script_index_order_field : string
    Order field for the scripts index page.
- enhancement_index_archived_filter : string
    Archived filter for the enhancements index page.
- project_index_order_dir : string
    Order direction for the projects index page.
- import_index_order_field : string
    Order field for the imports index page.
- result_index_author_filter : string
    Author filter for the results index page.
- script_index_archived_filter : string
    Archived filter for the scripts index page.
- import_index_author_filter : string
    Author filter for the imports index page.
- model_index_archived_filter : string
    Archived filter for the models index page.
- import_index_status_filter : string
    Status filter for the imports index page.
- enhancement_index_order_dir : string
    Order direction for the enhancements index page.
- result_index_order_field : string
    Order field for the results index page.
- report_index_thumbnail_view : string
    Thumbnail view for the reports index page.
- app_index_order_dir : string
    Order direction for the apps index pages.
- app_index_order_field : string
    Order field for the apps index pages.
- project_index_author_filter : string
    Author filter for the projects index page.
- project_index_order_field : string
    Order field for the projects index page.
- import_index_type_filter : string
    Type filter for the imports index page.
- model_index_author_filter : string
    Author filter for the models index page.
- enhancement_index_order_field : string
    Order field for the enhancements index page.
- script_index_type_filter : string
    Type filter for the scripts index page.

```

```
- project_detail_type_filter : string
    Type filter for projects detail pages.
- import_index_order_dir : string
    Order direction for the imports index page.
- script_index_author_filter : string
    Author filter for the scripts index page.
- export_index_order_dir : string
    Order direction for the exports index page.
- project_detail_order_dir : string
    Order direction for projects detail pages.
- project_detail_author_filter : string
    Author filter for projects detail pages.
- script_index_status_filter : string
    Status filter for the scripts index page.
- import_index_dest_filter : string
    Destination filter for the imports index page.
- model_index_status_filter : string
    Status filter for the models index page.
- result_index_type_filter : string
    Type filter for the results index page.
- result_index_order_dir : string
    Order direction for the results index page.
- preferred_server_id : integer
    ID of preferred server.
- civis_explore_skip_intro : boolean
    Whether the user is shown steps for each exploration.
- export_index_type_filter : string
    Type filter for the exports index page.
- model_index_order_field : string
    Order field for the models index page.
- script_index_order_dir : string
    Order direction for the scripts index page.
- enhancement_index_author_filter : string
    Author filter for the enhancements index page.
- import_index_archived_filter : string
    Archived filter for the imports index page.
```

Returns `username` : string

This user's username.

preferences : dict

This user's preferences.

organization_name : string

The name of the organization the user belongs to.

id : integer

The ID of this user.

feature_flags : dict

The feature flag settings for this user.

initials : string

This user's initials.

custom_branding : string

The branding of Platform for this user.

roles : list

The roles this user has, listed by slug.

email : string

This user's email address.

last_checked_announcements : string/date-time

The date and time at which the user last checked their announcements.

groups : list:

```
An array of all the groups this user is in.
- organization_id : integer
  The organization associated with this group.
- id : integer
  The ID of this group.
- name : string
  The name of this group.
```

name : string

This user's name.

post_api_keys (*id*, *expires_in*, *name*, ***kwargs*)

Create a new API key belonging to the logged-in user

Parameters *id* : string

The ID of the user or 'me'.

expires_in : integer

The number of seconds the key should last for.

name : string

The name of the API key.

constraints : list, optional:

```
Constraints on the abilities of the created key.
- post_allowed : boolean
  Whether the constraint allows POST requests.
- head_allowed : boolean
  Whether the constraint allows HEAD requests.
- delete_allowed : boolean
  Whether the constraint allows DELETE requests.
- put_allowed : boolean
  Whether the constraint allows PUT requests.
- 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.
- patch_allowed : boolean
  Whether the constraint allows PATCH requests.
```

Returns *expires_at* : string/date-time

The date and time when the key expired.

revoked_at : string/date-time

The date and time when the key was revoked.

token : string

The API key.

constraints : list:

Constraints on the abilities of the created key

- **post_allowed** : boolean
Whether the constraint allows POST requests.
- **head_allowed** : boolean
Whether the constraint allows HEAD requests.
- **delete_allowed** : boolean
Whether the constraint allows DELETE requests.
- **put_allowed** : boolean
Whether the constraint allows PUT requests.
- **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.
- **patch_allowed** : boolean
Whether the constraint allows PATCH requests.

use_count : integer

The number of times the key has been used.

scopes : list

The scopes which the key is permissioned on.

created_at : string/date-time

The date and time when the key was created.

expired : boolean

True if the key has expired.

last_used_at : string/date-time

The date and time when the key was last used.

active : boolean

True if the key has neither expired nor been revoked.

id : integer

The ID of the API key.

name : string

The name of the API key.

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.

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

C

`civis.parallel`, [41](#)

A

add_done_callback() (civis.ml.ModelFuture method), 37
 APIClient (class in civis), 46

C

cancel() (civis.ml.ModelFuture method), 37
 cancelled() (civis.ml.ModelFuture method), 37
 civis.parallel (module), 41
 CIVIS_API_KEY, 14–16, 18, 19, 21–27, 31, 33, 36, 46, 50
 civis_to_csv() (in module civis.io), 13
 civis_to_file() (in module civis.io), 22
 civis_to_multifile_csv() (in module civis.io), 15
 CivisFuture (class in civis.futures), 50
 Credentials (class in civis.resources._resources), 51
 csv_to_civis() (in module civis.io), 16

D

Databases (class in civis.resources._resources), 56
 dataframe_to_civis() (in module civis.io), 18
 default_credential (civis.APIClient attribute), 47
 delete_api_keys() (civis.resources._resources.Users method), 420
 delete_builds() (civis.resources._resources.Models method), 107
 delete_containers_projects() (civis.resources._resources.Scripts method), 193
 delete_containers_runs() (civis.resources._resources.Scripts method), 193
 delete_containers_shares_groups() (civis.resources._resources.Scripts method), 193
 delete_containers_shares_users() (civis.resources._resources.Scripts method), 193
 delete_custom_projects() (civis.resources._resources.Scripts method), 193

delete_custom_runs() (civis.resources._resources.Scripts method), 194
 delete_custom_shares_groups() (civis.resources._resources.Scripts method), 194
 delete_custom_shares_users() (civis.resources._resources.Scripts method), 194
 delete_files_runs() (civis.resources._resources.Imports method), 66
 delete_grants() (civis.resources._resources.Reports method), 168
 delete_javascript_projects() (civis.resources._resources.Scripts method), 194
 delete_javascript_runs() (civis.resources._resources.Scripts method), 194
 delete_javascript_shares_groups() (civis.resources._resources.Scripts method), 195
 delete_javascript_shares_users() (civis.resources._resources.Scripts method), 195
 delete_projects() (civis.resources._resources.Files method), 60
 delete_projects() (civis.resources._resources.Imports method), 66
 delete_projects() (civis.resources._resources.Jobs method), 98
 delete_projects() (civis.resources._resources.Models method), 107
 delete_projects() (civis.resources._resources.Reports method), 168
 delete_python3_projects() (civis.resources._resources.Scripts method), 195
 delete_python3_runs() (civis.resources._resources.Scripts method), 195
 delete_python3_shares_groups() (civis.resources._resources.Scripts method),

195
delete_python3_shares_users()
 (civis.resources._resources.Scripts method),
 196
delete_r_projects() (civis.resources._resources.Scripts
 method), 196
delete_r_runs() (civis.resources._resources.Scripts
 method), 196
delete_r_shares_groups()
 (civis.resources._resources.Scripts method),
 196
delete_r_shares_users() (civis.resources._resources.Scripts
 method), 197
delete_runs() (civis.resources._resources.Predictions
 method), 136
delete_runs() (civis.resources._resources.Queries
 method), 160
delete_shares_groups() (civis.resources._resources.Files
 method), 60
delete_shares_groups() (civis.resources._resources.Imports
 method), 67
delete_shares_groups() (civis.resources._resources.Jobs
 method), 98
delete_shares_groups() (civis.resources._resources.Models
 method), 108
delete_shares_groups() (civis.resources._resources.Projects
 method), 145
delete_shares_groups() (civis.resources._resources.Reports
 method), 169
delete_shares_users() (civis.resources._resources.Files
 method), 61
delete_shares_users() (civis.resources._resources.Imports
 method), 67
delete_shares_users() (civis.resources._resources.Jobs
 method), 98
delete_shares_users() (civis.resources._resources.Models
 method), 108
delete_shares_users() (civis.resources._resources.Projects
 method), 145
delete_shares_users() (civis.resources._resources.Reports
 method), 169
delete_sql_projects() (civis.resources._resources.Scripts
 method), 197
delete_sql_runs() (civis.resources._resources.Scripts
 method), 197
delete_sql_shares_groups()
 (civis.resources._resources.Scripts method),
 197
delete_sql_shares_users()
 (civis.resources._resources.Scripts method),
 197
delete_syncs() (civis.resources._resources.Imports
 method), 67
delete_whitelist_ips() (civis.resources._resources.Databases

 method), 58
done() (civis.ml.ModelFuture method), 37

E

environment variable
 CIVIS_API_KEY, 14–16, 18, 19, 21–27, 31, 33, 36,
 46, 50
exception() (civis.ml.ModelFuture method), 37

F

failed() (civis.ml.ModelFuture method), 38
file_id_from_run_output() (in module civis.io), 23
file_to_civis() (in module civis.io), 23
file_to_dataframe() (in module civis.io), 24
file_to_json() (in module civis.io), 25
Files (class in civis.resources._resources), 60
from_existing() (civis.ml.ModelPipeline class method),
 33

G

get() (civis.resources._resources.Credentials method), 51
get() (civis.resources._resources.Files method), 61
get() (civis.resources._resources.Imports method), 67
get() (civis.resources._resources.Jobs method), 99
get() (civis.resources._resources.Models method), 108
get() (civis.resources._resources.Predictions method),
 136
get() (civis.resources._resources.Projects method), 145
get() (civis.resources._resources.Queries method), 160
get() (civis.resources._resources.Reports method), 169
get() (civis.resources._resources.Scripts method), 198
get() (civis.resources._resources.Tables method), 402
get() (civis.resources._resources.Users method), 421
get_api_keys() (civis.resources._resources.Users
 method), 422
get_aws_credential_id() (civis.APIClient method), 47
get_batches() (civis.resources._resources.Imports
 method), 70
get_builds() (civis.resources._resources.Models method),
 112
get_containers() (civis.resources._resources.Scripts
 method), 201
get_containers_runs() (civis.resources._resources.Scripts
 method), 205
get_custom() (civis.resources._resources.Scripts
 method), 206
get_custom_runs() (civis.resources._resources.Scripts
 method), 209
get_database_credential_id() (civis.APIClient method),
 48
get_database_id() (civis.APIClient method), 48
get_enhancements_cass_ncoa()
 (civis.resources._resources.Tables method),
 405

- `get_enhancements_geocodings()`
(`civis.resources._resources.Tables` method), 406
- `get_enhancements_prepared_matchings()`
(`civis.resources._resources.Tables` method), 407
- `get_enhancements_table_matchings()`
(`civis.resources._resources.Tables` method), 407
- `get_files_runs()` (`civis.resources._resources.Imports` method), 71
- `get_javascript()` (`civis.resources._resources.Scripts` method), 209
- `get_javascript_runs()` (`civis.resources._resources.Scripts` method), 213
- `get_python3()` (`civis.resources._resources.Scripts` method), 213
- `get_python3_runs()` (`civis.resources._resources.Scripts` method), 217
- `get_r()` (`civis.resources._resources.Scripts` method), 217
- `get_r_runs()` (`civis.resources._resources.Scripts` method), 221
- `get_runs()` (`civis.resources._resources.Jobs` method), 100
- `get_runs()` (`civis.resources._resources.Predictions` method), 138
- `get_runs()` (`civis.resources._resources.Queries` method), 161
- `get_sql()` (`civis.resources._resources.Scripts` method), 222
- `get_sql_runs()` (`civis.resources._resources.Scripts` method), 226
- `get_table_id()` (`civis.APIClient` method), 49
- `get_whitelist_ips()` (`civis.resources._resources.Databases` method), 58
- I**
- `Imports` (class in `civis.resources._resources`), 66
- `infer_backend_factory()` (in module `civis.parallel`), 41
- J**
- `Jobs` (class in `civis.resources._resources`), 98
- L**
- `list()` (`civis.resources._resources.Credentials` method), 52
- `list()` (`civis.resources._resources.Databases` method), 58
- `list()` (`civis.resources._resources.Imports` method), 72
- `list()` (`civis.resources._resources.Jobs` method), 100
- `list()` (`civis.resources._resources.Models` method), 113
- `list()` (`civis.resources._resources.Predictions` method), 139
- `list()` (`civis.resources._resources.Projects` method), 148
- `list()` (`civis.resources._resources.Queries` method), 162
- `list()` (`civis.resources._resources.Reports` method), 171
- `list()` (`civis.resources._resources.Scripts` method), 226
- `list()` (`civis.resources._resources.Tables` method), 408
- `list()` (`civis.resources._resources.Users` method), 423
- `list_api_keys()` (`civis.resources._resources.Users` method), 424
- `list_batches()` (`civis.resources._resources.Imports` method), 74
- `list_builds()` (`civis.resources._resources.Models` method), 116
- `list_children()` (`civis.resources._resources.Jobs` method), 101
- `list_columns()` (`civis.resources._resources.Tables` method), 410
- `list_containers_projects()`
(`civis.resources._resources.Scripts` method), 228
- `list_containers_runs()` (`civis.resources._resources.Scripts` method), 229
- `list_containers_runs_logs()`
(`civis.resources._resources.Scripts` method), 230
- `list_containers_runs_outputs()`
(`civis.resources._resources.Scripts` method), 231
- `list_containers_shares()` (`civis.resources._resources.Scripts` method), 231
- `list_custom()` (`civis.resources._resources.Scripts` method), 232
- `list_custom_projects()` (`civis.resources._resources.Scripts` method), 234
- `list_custom_runs()` (`civis.resources._resources.Scripts` method), 235
- `list_custom_runs_logs()` (`civis.resources._resources.Scripts` method), 236
- `list_custom_runs_outputs()`
(`civis.resources._resources.Scripts` method), 236
- `list_custom_shares()` (`civis.resources._resources.Scripts` method), 237
- `list_files_runs()` (`civis.resources._resources.Imports` method), 75
- `list_history()` (`civis.resources._resources.Scripts` method), 238
- `list_javascript_projects()` (`civis.resources._resources.Scripts` method), 238
- `list_javascript_runs()` (`civis.resources._resources.Scripts` method), 239
- `list_javascript_runs_logs()`
(`civis.resources._resources.Scripts` method), 240
- `list_javascript_runs_outputs()`
(`civis.resources._resources.Scripts` method), 240
- `list_javascript_shares()` (`civis.resources._resources.Scripts` method), 241

`list_me()` (civis.resources._resources.Users method), [425](#)
`list_parents()` (civis.resources._resources.Jobs method), [102](#)
`list_projects()` (civis.resources._resources.Files method), [61](#)
`list_projects()` (civis.resources._resources.Imports method), [76](#)
`list_projects()` (civis.resources._resources.Jobs method), [103](#)
`list_projects()` (civis.resources._resources.Models method), [118](#)
`list_projects()` (civis.resources._resources.Reports method), [173](#)
`list_python3_projects()` (civis.resources._resources.Scripts method), [242](#)
`list_python3_runs()` (civis.resources._resources.Scripts method), [243](#)
`list_python3_runs_logs()` (civis.resources._resources.Scripts method), [244](#)
`list_python3_runs_outputs()` (civis.resources._resources.Scripts method), [244](#)
`list_python3_shares()` (civis.resources._resources.Scripts method), [245](#)
`list_r_projects()` (civis.resources._resources.Scripts method), [246](#)
`list_r_runs()` (civis.resources._resources.Scripts method), [247](#)
`list_r_runs_logs()` (civis.resources._resources.Scripts method), [247](#)
`list_r_runs_outputs()` (civis.resources._resources.Scripts method), [248](#)
`list_r_shares()` (civis.resources._resources.Scripts method), [249](#)
`list_runs()` (civis.resources._resources.Imports method), [76](#)
`list_runs()` (civis.resources._resources.Predictions method), [139](#)
`list_runs()` (civis.resources._resources.Queries method), [163](#)
`list_schedules()` (civis.resources._resources.Models method), [118](#)
`list_schedules()` (civis.resources._resources.Predictions method), [140](#)
`list_schemas()` (civis.resources._resources.Databases method), [59](#)
`list_shares()` (civis.resources._resources.Files method), [62](#)
`list_shares()` (civis.resources._resources.Imports method), [77](#)
`list_shares()` (civis.resources._resources.Jobs method), [103](#)
`list_shares()` (civis.resources._resources.Models method), [119](#)
`list_shares()` (civis.resources._resources.Projects method), [149](#)
`list_shares()` (civis.resources._resources.Reports method), [174](#)
`list_snapshots()` (civis.resources._resources.Reports method), [175](#)
`list_sql_projects()` (civis.resources._resources.Scripts method), [249](#)
`list_sql_runs()` (civis.resources._resources.Scripts method), [250](#)
`list_sql_runs_logs()` (civis.resources._resources.Scripts method), [251](#)
`list_sql_runs_outputs()` (civis.resources._resources.Scripts method), [252](#)
`list_sql_shares()` (civis.resources._resources.Scripts method), [252](#)
`list_types()` (civis.resources._resources.Models method), [120](#)
`list_types()` (civis.resources._resources.Scripts method), [253](#)
`list_whitelist_ips()` (civis.resources._resources.Databases method), [59](#)

M

`make_backend_factory()` (in module `civis.parallel`), [43](#)
`make_backend_template_factory()` (in module `civis.parallel`), [45](#)
`ModelFuture` (class in `civis.ml`), [35](#)
`ModelPipeline` (class in `civis.ml`), [30](#)
`Models` (class in `civis.resources._resources`), [107](#)

P

`PaginatedResponse` (class in `civis.response`), [49](#)
`patch()` (civis.resources._resources.Models method), [120](#)
`patch()` (civis.resources._resources.Predictions method), [141](#)
`patch()` (civis.resources._resources.Reports method), [176](#)
`patch()` (civis.resources._resources.Scripts method), [253](#)
`patch()` (civis.resources._resources.Tables method), [411](#)
`patch_containers()` (civis.resources._resources.Scripts method), [258](#)
`patch_containers_runs()` (civis.resources._resources.Scripts method), [264](#)
`patch_custom()` (civis.resources._resources.Scripts method), [265](#)
`patch_javascript()` (civis.resources._resources.Scripts method), [269](#)
`patch_me()` (civis.resources._resources.Users method), [426](#)
`patch_python3()` (civis.resources._resources.Scripts method), [274](#)
`patch_r()` (civis.resources._resources.Scripts method), [280](#)

[patch_snapshots\(\)](#) (civis.resources._resources.Reports method), [178](#)
[patch_sql\(\)](#) (civis.resources._resources.Scripts method), [285](#)
[post\(\)](#) (civis.resources._resources.Credentials method), [53](#)
[post\(\)](#) (civis.resources._resources.Files method), [63](#)
[post\(\)](#) (civis.resources._resources.Imports method), [77](#)
[post\(\)](#) (civis.resources._resources.Models method), [122](#)
[post\(\)](#) (civis.resources._resources.Projects method), [150](#)
[post\(\)](#) (civis.resources._resources.Queries method), [164](#)
[post\(\)](#) (civis.resources._resources.Reports method), [180](#)
[post\(\)](#) (civis.resources._resources.Scripts method), [291](#)
[post\(\)](#) (civis.resources._resources.Tables method), [413](#)
[post_api_keys\(\)](#) (civis.resources._resources.Users method), [429](#)
[post_authenticate\(\)](#) (civis.resources._resources.Credentials method), [54](#)
[post_batches\(\)](#) (civis.resources._resources.Imports method), [82](#)
[post_builds\(\)](#) (civis.resources._resources.Models method), [127](#)
[post_cancel\(\)](#) (civis.resources._resources.Imports method), [83](#)
[post_cancel\(\)](#) (civis.resources._resources.Scripts method), [296](#)
[post_containers\(\)](#) (civis.resources._resources.Scripts method), [296](#)
[post_containers_runs\(\)](#) (civis.resources._resources.Scripts method), [302](#)
[post_containers_runs_heartbeats\(\)](#) (civis.resources._resources.Scripts method), [303](#)
[post_containers_runs_logs\(\)](#) (civis.resources._resources.Scripts method), [303](#)
[post_containers_runs_outputs\(\)](#) (civis.resources._resources.Scripts method), [304](#)
[post_custom\(\)](#) (civis.resources._resources.Scripts method), [304](#)
[post_custom_runs\(\)](#) (civis.resources._resources.Scripts method), [308](#)
[post_custom_runs_outputs\(\)](#) (civis.resources._resources.Scripts method), [309](#)
[post_enhancements_cass_ncoa\(\)](#) (civis.resources._resources.Tables method), [413](#)
[post_enhancements_geocodings\(\)](#) (civis.resources._resources.Tables method), [414](#)
[post_enhancements_prepared_matchings\(\)](#) (civis.resources._resources.Tables method), [414](#)
[post_enhancements_table_matchings\(\)](#) (civis.resources._resources.Tables method), [415](#)
[post_files\(\)](#) (civis.resources._resources.Imports method), [83](#)
[post_files_runs\(\)](#) (civis.resources._resources.Imports method), [84](#)
[post_grants\(\)](#) (civis.resources._resources.Reports method), [182](#)
[post_javascript\(\)](#) (civis.resources._resources.Scripts method), [309](#)
[post_javascript_runs\(\)](#) (civis.resources._resources.Scripts method), [314](#)
[post_javascript_runs_outputs\(\)](#) (civis.resources._resources.Scripts method), [315](#)
[post_python3\(\)](#) (civis.resources._resources.Scripts method), [315](#)
[post_python3_runs\(\)](#) (civis.resources._resources.Scripts method), [321](#)
[post_python3_runs_outputs\(\)](#) (civis.resources._resources.Scripts method), [321](#)
[post_r\(\)](#) (civis.resources._resources.Scripts method), [322](#)
[post_r_runs\(\)](#) (civis.resources._resources.Scripts method), [327](#)
[post_r_runs_outputs\(\)](#) (civis.resources._resources.Scripts method), [328](#)
[post_refresh\(\)](#) (civis.resources._resources.Tables method), [416](#)
[post_run\(\)](#) (civis.resources._resources.Scripts method), [328](#)
[post_runs\(\)](#) (civis.resources._resources.Imports method), [85](#)
[post_runs\(\)](#) (civis.resources._resources.Jobs method), [104](#)
[post_runs\(\)](#) (civis.resources._resources.Predictions method), [143](#)
[post_runs\(\)](#) (civis.resources._resources.Queries method), [166](#)
[post_snapshots\(\)](#) (civis.resources._resources.Reports method), [184](#)
[post_sql\(\)](#) (civis.resources._resources.Scripts method), [329](#)
[post_sql_runs\(\)](#) (civis.resources._resources.Scripts method), [335](#)
[post_syncs\(\)](#) (civis.resources._resources.Imports method), [85](#)
[post_temporary\(\)](#) (civis.resources._resources.Credentials method), [55](#)
[post_trigger_email\(\)](#) (civis.resources._resources.Jobs method), [105](#)
[post_whitelist_ips\(\)](#) (civis.resources._resources.Databases

method), 59
predict() (civis.ml.ModelPipeline method), 33
Predictions (class in civis.resources._resources), 136
Projects (class in civis.resources._resources), 144
put() (civis.resources._resources.Credentials method), 55
put() (civis.resources._resources.Imports method), 87
put() (civis.resources._resources.Projects method), 152
put_archive() (civis.resources._resources.Imports method), 91
put_archive() (civis.resources._resources.Models method), 128
put_archive() (civis.resources._resources.Projects method), 155
put_archive() (civis.resources._resources.Reports method), 185
put_containers() (civis.resources._resources.Scripts method), 335
put_containers_archive() (civis.resources._resources.Scripts method), 342
put_containers_projects() (civis.resources._resources.Scripts method), 346
put_containers_shares_groups() (civis.resources._resources.Scripts method), 346
put_containers_shares_users() (civis.resources._resources.Scripts method), 347
put_custom() (civis.resources._resources.Scripts method), 348
put_custom_archive() (civis.resources._resources.Scripts method), 352
put_custom_projects() (civis.resources._resources.Scripts method), 355
put_custom_shares_groups() (civis.resources._resources.Scripts method), 355
put_custom_shares_users() (civis.resources._resources.Scripts method), 356
put_javascript() (civis.resources._resources.Scripts method), 357
put_javascript_archive() (civis.resources._resources.Scripts method), 362
put_javascript_projects() (civis.resources._resources.Scripts method), 365
put_javascript_shares_groups() (civis.resources._resources.Scripts method), 366
put_javascript_shares_users() (civis.resources._resources.Scripts method), 366
put_predictions() (civis.resources._resources.Models method), 132
put_projects() (civis.resources._resources.Files method), 64
put_projects() (civis.resources._resources.Imports method), 94
put_projects() (civis.resources._resources.Jobs method), 105
put_projects() (civis.resources._resources.Models method), 133
put_projects() (civis.resources._resources.Reports method), 187
put_python3() (civis.resources._resources.Scripts method), 367
put_python3_archive() (civis.resources._resources.Scripts method), 373
put_python3_projects() (civis.resources._resources.Scripts method), 376
put_python3_shares_groups() (civis.resources._resources.Scripts method), 377
put_python3_shares_users() (civis.resources._resources.Scripts method), 377
put_r() (civis.resources._resources.Scripts method), 378
put_r_archive() (civis.resources._resources.Scripts method), 384
put_r_projects() (civis.resources._resources.Scripts method), 388
put_r_shares_groups() (civis.resources._resources.Scripts method), 388
put_r_shares_users() (civis.resources._resources.Scripts method), 389
put_schedules() (civis.resources._resources.Models method), 133
put_schedules() (civis.resources._resources.Predictions method), 143
put_scripts() (civis.resources._resources.Queries method), 166
put_shares_groups() (civis.resources._resources.Files method), 64
put_shares_groups() (civis.resources._resources.Imports method), 94
put_shares_groups() (civis.resources._resources.Jobs method), 105
put_shares_groups() (civis.resources._resources.Models method), 134
put_shares_groups() (civis.resources._resources.Projects method), 158
put_shares_groups() (civis.resources._resources.Reports method), 188
put_shares_users() (civis.resources._resources.Files method), 65
put_shares_users() (civis.resources._resources.Imports

method), 95
 put_shares_users() (civis.resources._resources.Jobs
 method), 106
 put_shares_users() (civis.resources._resources.Models
 method), 135
 put_shares_users() (civis.resources._resources.Projects
 method), 159
 put_shares_users() (civis.resources._resources.Reports
 method), 188
 put_sql() (civis.resources._resources.Scripts method),
 390
 put_sql_archive() (civis.resources._resources.Scripts
 method), 396
 put_sql_projects() (civis.resources._resources.Scripts
 method), 400
 put_sql_shares_groups() (civis.resources._resources.Scripts
 method), 400
 put_sql_shares_users() (civis.resources._resources.Scripts
 method), 401
 put_syncs() (civis.resources._resources.Imports method),
 96

Q

Queries (class in civis.resources._resources), 160
 query_civis() (in module civis.io), 26

R

read_civis() (in module civis.io), 19
 read_civis_sql() (in module civis.io), 20
 Reports (class in civis.resources._resources), 168
 Response (class in civis.response), 49
 result() (civis.ml.ModelFuture method), 38
 running() (civis.ml.ModelFuture method), 38

S

Scripts (class in civis.resources._resources), 189
 set_exception() (civis.ml.ModelFuture method), 38
 set_result() (civis.ml.ModelFuture method), 38
 set_running_or_notify_cancel() (civis.ml.ModelFuture
 method), 38
 succeeded() (civis.ml.ModelFuture method), 38

T

Tables (class in civis.resources._resources), 402
 train() (civis.ml.ModelPipeline method), 34
 transfer_table() (in module civis.io), 25

U

username (civis.APIClient attribute), 49
 Users (class in civis.resources._resources), 420