
Civis Client Documentation

Release 1.0.0

Civis Analytics

Mar 24, 2017

Contents

1	Installation	3
2	Authentication	5
3	User Guide	7
4	Client API Reference	9
5	Indices and tables	365

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
```


CHAPTER 2

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:

```
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:

```
CIVIS_API_KEY="api_key_here"
```


CHAPTER 3

User Guide

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

User Guide

Getting Started

After installing the Civi 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 Civi 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 Civi, calculates the correlation between all the columns and then uploads the data back into Civi:

```
>>> 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
>>> poller = civis.io.dataframe_to_civis(df=correlation_matrix,
...                                     database="database",
...                                     table="my_schema.my_correlations")
...
>>> poller.result()
```

Pollable Results

In the code above, `dataframe_to_civis()` returns a special `PollableResult` object. Making a request to the Civi API usually results in a long running job. To account for this, various functions in the `civis` namespace return a `PollableResult` 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 `PollableResult` follows the `concurrent.futures.Future` API fairly closely. For example, calling `result()` on `poller` 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>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, credential_id=None, polling_interval=15, archive=True)
```

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 : str, optional

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

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 query completion.

archive : bool, optional

If `True` (the default), archive the export job as soon as it completes.

Returns results : *PollableResult*

A *PollableResult* 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"
>>> poll = civis_to_csv("file.csv", sql, "my_database")
>>> poll.result() # Wait for job to complete
```

civis.io.csv_to_civis

civis.io.csv_to_civis (*filename*, *database*, *table*, *api_key*=*None*, *max_errors*=*None*, *existing_table_rows*=*'fail'*, *distkey*=*None*, *sortkey1*=*None*, *sortkey2*=*None*, *delimiter*=*'*, *headers*=*None*, *credential_id*=*None*, *polling_interval*=*15*, *archive*=*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 : str, optional

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

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

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

Returns **results** : `PollableResult`

A `PollableResult` 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')
>>> poller = civis.io.csv_to_civis('input_file.csv',
...                               'my-database',
...                               'scratch.my_data')
>>> poller.result()
```

civis.io.dataframe_to_civis

`civis.io.dataframe_to_civis(df, database, table, api_key=None, max_errors=None, existing_table_rows='fail', distkey=None, sortkey1=None, sortkey2=None, headers=None, credential_id=None, polling_interval=15, archive=True, **kwargs)`

Upload a *pandas DataFrame* into a Civis table.

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 : str, optional

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

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

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

****kwargs** : kwargs

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

Returns `poll`: `PollableResult`

A *PollableResult* object.

Examples

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

civis.io.read_civis

`civis.io.read_civis`(*table*, *database*, *columns=None*, *use_pandas=False*, *job_name=None*, *api_key=None*, *credential_id=None*, *polling_interval=15*, *archive=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 : str, optional

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

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

If True (the default), archive the export job as soon as it completes.

***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*, *credential_id=None*, *polling_interval=15*, *archive=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 : str, optional

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

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

If `True` (the default), archive the export job as soon as it completes.

****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])</code>	Download a file from Civis.
<code>file_to_civis(buf, name[, api_key])</code>	Upload a file to Civis.

`civis.io.civis_to_file`

`civis.io.civis_to_file` (*file_id*, *buf*, *api_key=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 : str, optional

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

Returns None

Examples

```
>>> file_id = 100
>>> with open("my_file.txt", "w") as f:
...     civis_to_file(file_id, f)
```

civis.io.file_to_civis

`civis.io.file_to_civis(buf, name, api_key=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 : str, optional

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

****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')`).

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, source_credential_id=None, dest_credential_id=None,
                        polling_interval=15, **advanced_options)
```

Transfer a table from one location to another.

Parameters **source_db** : str or int

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

dest_db : str or int

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

source_table : str

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

dest_table : str

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

job_name : str, optional

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

api_key : str, optional

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

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** : [PollableResult](#)

A [PollableResult](#) 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, credential_id=None, preview_rows=10,
                    polling_interval=15)
```

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 : str, optional

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

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.

Returns `results` : *PollableResult*

A *PollableResult* object.

Examples

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

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)
```

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

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.

Attributes

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

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

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

A response object 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.polling.PollableResult* (*poller, poller_args, polling_interval=15*)

Bases: *concurrent.futures._base.Future*

A class for tracking pollable results.

This class will begin polling immediately upon creation, and poll for job completion once every *polling_interval* seconds until the job completes in Civis.

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

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

cancel ()

Not currently implemented.

failed ()

Return *True* if the Civis job failed.

succeeded ()

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

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

get (*id*)

Get a credential

Parameters *id* : integer

The ID of the credential.

Returns *id* : integer

The ID of the credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

owner : string

The name of the user who this credential belongs to.

description : string

A long description of the credential.

remote_host_name : string

The name of the remote host associated with this credential.

type : string

The credential's type.

updated_at : string/time

The last modification time for this credential.

name : string

The name identifying the credential

username : string

The username for the credential.

created_at : string/time

The creation time for this 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").

Returns id : integer

The ID of the credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

owner : string

The name of the user who this credential belongs to.

description : string

A long description of the credential.

remote_host_name : string

The name of the remote host associated with this credential.

type : string

The credential's type.

updated_at : string/time

The last modification time for this credential.

name : string

The name identifying the credential

username : string

The username for the credential.

created_at : string/time

The creation time for this credential.

post (*password, username, type, **kwargs*)
Create or update a credential

Parameters password : string

The password for the credential.

username : string

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

The ID of the credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

owner : string

The name of the user who this credential belongs to.

description : string

A long description of the credential.

remote_host_name : string

The name of the remote host associated with this credential.

type : string

The credential's type.

updated_at : string/time

The last modification time for this credential.

name : string

The name identifying the credential

username : string

The username for the credential.

created_at : string/time

The creation time for this credential.

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

Authenticate against a remote host

Parameters username : string

The username for the credential.

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

password : string

The password for the credential.

url : string

The URL to your host.

Returns **id** : integer

The ID of the credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

owner : string

The name of the user who this credential belongs to.

description : string

A long description of the credential.

remote_host_name : string

The name of the remote host associated with this credential.

type : string

The credential's type.

updated_at : string/time

The last modification time for this credential.

name : string

The name identifying the credential

username : string

The username for the credential.

created_at : string/time

The creation time for this 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.

access_key : string

The identifier of the credential.

secret_access_key : string

The secret part of the credential.

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

Update an existing credential

Parameters **id** : integer

The ID of the credential.

password : string

The password for the credential.

username : string

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

The ID of the credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

owner : string

The name of the user who this credential belongs to.

description : string

A long description of the credential.

remote_host_name : string

The name of the remote host associated with this credential.

type : string

The credential's type.

updated_at : string/time

The last modification time for this credential.

name : string

The name identifying the credential

username : string

The username for the credential.

created_at : string/time

The creation time for this 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 **authorized_by** : string

The user who authorized this rule.

subnet_mask : string

The subnet mask that is allowed by this rule.

created_at : string/time

The time this rule was created.

security_group_id : string

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

is_active : boolean

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

updated_at : string/time

The time this rule was last updated.

id : integer

The ID of this whitelisted IP address.

remote_host_id : integer

The ID of the database this rule is applied to.

list ()

List databases

Returns id : integer

The ID for the database.

name : string

The name of the database.

list_schemas (*id*)

List schemas in this database

Parameters id : integer

The ID of the database.

Returns schema : string

The name of a schema.

list_whitelist_ips (*id*)

List whitelisted IPs for the specified database

Parameters id : integer

The ID for the database.

Returns id : integer

The ID of this whitelisted IP address.

subnet_mask : string

The subnet mask that is allowed by this rule.

created_at : string/time

The time this rule was created.

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.

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 *authorized_by* : string

The user who authorized this rule.

subnet_mask : string

The subnet mask that is allowed by this rule.

created_at : string/time

The time this rule was created.

security_group_id : string

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

is_active : boolean

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

updated_at : string/time

The time this rule was last updated.

id : integer

The ID of this whitelisted IP address.

remote_host_id : integer

The ID of the database this rule is applied to.

Files

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

Methods

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

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

id : integer

The ID of the file object.

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.

created_at : string/date-time

The date and time the file was created.

file_url : string

The URL that may be used to download the file.

name : string

The file name.

file_size : integer

The file size.

list_projects (*id*)

List the projects a Data::S3File belongs to

Parameters **id** : integer

The ID of the resource.

Returns **id** : integer

The ID for this project.

archived : string

The archival status of the requested object(s).

author : dict:

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

created_at : string/time

description : string

A description of the project

updated_at : string/time

name : string

The name of this project.

auto_share : boolean

users : list:

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

list_shares (*id*)

List users and groups permissioned on this object

Parameters *id* : integer

The ID of the object.

Returns *readers* : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

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 *id* : integer

The ID of the file object.

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.

created_at : string/date-time

The date and time the file was created.

upload_fields : dict

A hash containing the form fields to be included with the POST request.

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.

name : string

The file name.

file_size : integer

The file size.

put_projects (*id*, *project_id*)

Add a Data::S3File to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

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

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

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

Returns **readers** : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

put_shares_users (*id*, *permission_level*, *user_ids*)
Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

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

user_ids : list

An array of one or more user IDs

Returns **readers** : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

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_files_runs</i> (id, run_id)	Check status of a run
<i>list</i> (**kwargs)	List 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> (sync_type, name, is_outbound, **kwargs)	Create a new import configuration
<i>post_files</i> (schema, remote_host_id, ...)	Initiate 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, destination, source, **kwargs)	Create a sync
<i>put</i> (id, sync_type, name, is_outbound, **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, group_ids, ...)	Set the permissions groups has on this object
<i>put_shares_users</i> (id, permission_level, user_ids)	Set the permissions users have on this object
<i>put_syncs</i> (id, sync_id, destination, source, ...)	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 **archived** : string

The archival status of the requested object(s).

id : integer

The ID for the import.

running_as : dict:

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

schedule : dict:

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

syncs : list:

```
List of syncs.
- id : integer
- advanced_options : dict::
    - verify_table_row_counts : boolean
    - contact_lists : string
    - first_row_is_header : boolean
    - export_action : string
    - max_errors : integer
    - wipe_destination_table : boolean
    - distkey : string
    - partition_column_name : string
    - identity_column : string
    - last_modified_column : string
    - column_delimiter : string
    - soql_query : string
    - partition_table_partition_column_min_name : string
    - sortkey2 : string
    - partition_table_partition_column_max_name : string
    - sortkey1 : string
    - partition_table_name : string
    - existing_table_rows : string
    - truncate_long_lines : boolean
    - invalid_char_replacement : string
    - row_chunk_size : integer
    - mysql_catalog_matches_schema : boolean
```

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

last_run : dict:

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

hidden : boolean

The hidden status of the object.

notifications : dict:

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

updated_at : string/date-time

user : dict:

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

```

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

```

source : dict:

```

-   credential_id : integer
-   additional_credentials : list
        Array that holds additional credentials used for specific_
↳ imports. For
        salesforce imports, the first and only element is the client_
↳ credential
        id.
-   name : string
-   remote_host_id : integer

```

is_outbound : boolean

next_run_at : string/time

The time of the next scheduled run.

destination : dict:

```

-   credential_id : integer
-   additional_credentials : list
        Array that holds additional credentials used for specific_
↳ imports. For
        salesforce imports, the first and only element is the client_
↳ credential
        id.
-   name : string
-   remote_host_id : integer

```

parent_id : integer

Parent id to trigger this import from

sync_type : string

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

time_zone : string

The time zone of this import.

state : string

name : string

The name of the import.

created_at : string/date-time

get_files_runs (*id*, *run_id*)

Check status of a run

Parameters *id* : integer

The ID of the import.

run_id : integer

The ID of the run.

Returns **id** : integer

The ID of the run.

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.

started_at : string/time

The time the last run started at.

import_id : integer

The ID of the import.

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 id : integer

The ID for the import.

archived : string

The archival status of the requested object(s).

schedule : dict:

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

last_run : dict:

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

updated_at : string/date-time

user : dict:

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

source : dict:

```
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
↳ imports. For
    salesforce imports, the first and only element is the client_
↳ credential
    id.
- name : string
- remote_host_id : integer
```

is_outbound : boolean

created_at : string/date-time

sync_type : string

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

time_zone : string

The time zone of this import.

state : string

name : string

The name of the import.

destination : dict:

```
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
↳ imports. For
    salesforce imports, the first and only element is the client_
↳ credential
    id.
- name : string
- remote_host_id : integer
```

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 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.

started_at : string/time

The time the last run started at.

import_id : integer

The ID of the import.

list_projects (*id*)

List the projects a JobTypes::Import belongs to

Parameters id : integer

The ID of the resource.

Returns id : integer

The ID for this project.

archived : string

The archival status of the requested object(s).

author : dict:

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

created_at : string/time

description : string

A description of the project

updated_at : string/time

name : string

The name of this project.

auto_share : boolean

users : list:

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

list_runs (*id*)

Get the run history of this import

Parameters *id* : integer

Returns *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.

started_at : string/time

The time that the run started.

list_shares (*id*)

List users and groups permissioned on this object

Parameters *id* : integer

The ID of the object.

Returns *readers* : dict:

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

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

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

Create a new import configuration

Parameters **sync_type** : string

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

name : string

The name of the import.

is_outbound : boolean

hidden : boolean, optional

The hidden status of the object.

next_run_at : string/time, optional

The time of the next scheduled run.

schedule : dict, optional:

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

destination : dict, optional:

```
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
↪imports. For
    salesforce imports, the first and only element is the client_
↪credential
    id.
- remote_host_id : integer
```

parent_id : integer, optional

Parent id to trigger this import from

time_zone : string, optional

The time zone of this import.

notifications : dict, optional:

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

source : dict, optional:

```
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
↪imports. For
    salesforce imports, the first and only element is the client_
↪credential
    id.
- remote_host_id : integer
```

Returns archived : string

The archival status of the requested object(s).

id : integer

The ID for the import.

running_as : dict:

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

schedule : dict:

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

syncs : list:

```
List of syncs.
- id : integer
- advanced_options : dict::
    - verify_table_row_counts : boolean
    - contact_lists : string
    - first_row_is_header : boolean
    - export_action : string
    - max_errors : integer
    - wipe_destination_table : boolean
    - distkey : string
    - partition_column_name : string
    - identity_column : string
    - last_modified_column : string
    - column_delimiter : string
    - soql_query : string
    - partition_table_partition_column_min_name : string
    - sortkey2 : string
    - partition_table_partition_column_max_name : string
    - sortkey1 : string
    - partition_table_name : string
    - existing_table_rows : string
    - truncate_long_lines : boolean
    - invalid_char_replacement : string
    - row_chunk_size : integer
    - mysql_catalog_matches_schema : boolean
    - partition_schema_name : string
- destination : dict::
    - path : string
      The schema.tablename to sync to.
- source : dict::
```

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

last_run : dict:

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

hidden : boolean

The hidden status of the object.

notifications : dict:

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

updated_at : string/date-time

user : dict:

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

source : dict:


```

- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
    ↪ imports. For
    ↪ salesforce imports, the first and only element is the client_
    ↪ credential
    ↪ id.
- name : string
- remote_host_id : integer

```

is_outbound : boolean

next_run_at : string/time

The time of the next scheduled run.

destination : dict:

```

- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
    ↪ imports. For
    ↪ salesforce imports, the first and only element is the client_
    ↪ credential
    ↪ id.
- name : string
- remote_host_id : integer

```

parent_id : integer

Parent id to trigger this import from

sync_type : string

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

time_zone : string

The time zone of this import.

state : string

name : string

The name of the import.

created_at : string/date-time

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.

multipart : boolean, optional

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

sortkey1 : string, optional

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

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.

max_errors : integer, optional

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

existing_table_rows : string, optional

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

distkey : string, optional

The column to use as the distkey for the table.

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”.

sortkey2 : string, optional

The second column in a compound sortkey for the table.

Returns id : integer

The id of the import.

upload_fields : dict

If multipart was set to true, these fields should be included in the multipart upload.

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_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 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.

started_at : string/time

The time the last run started at.

import_id : integer

The ID of the import.

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, destination, source, **kwargs*)

Create a sync

Parameters **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.
```

advanced_options : dict, optional:

```
- verify_table_row_counts : boolean
- contact_lists : string
- first_row_is_header : boolean
- export_action : string
- max_errors : integer
- wipe_destination_table : boolean
- distkey : string
- partition_column_name : string
- identity_column : string
- last_modified_column : string
- column_delimiter : string
- soql_query : string
```

```
- partition_table_partition_column_min_name : string
- sortkey2 : string
- partition_table_partition_column_max_name : string
- sortkey1 : string
- partition_table_name : string
- existing_table_rows : string
- truncate_long_lines : boolean
- invalid_char_replacement : string
- row_chunk_size : integer
- mysql_catalog_matches_schema : boolean
- partition_schema_name : string
```

Returns `id` : integer

advanced_options : dict:

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

destination : dict:

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

source : dict:

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

put (*id*, *sync_type*, *name*, *is_outbound*, ***kwargs*)

Update an import

Parameters `id` : integer

The ID for the import.

sync_type : string

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

name : string

The name of the import.

is_outbound : boolean

next_run_at : string/time, optional

The time of the next scheduled run.

schedule : dict, optional:

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

destination : dict, optional:

```
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific_
↪imports. For
    salesforce imports, the first and only element is the client_
↪credential
    id.
- remote_host_id : integer
```

parent_id : integer, optional

Parent id to trigger this import from

time_zone : string, optional

The time zone of this import.

notifications : dict, optional:

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

```
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
↳ successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
```

source : dict, optional:

```
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific
↳ imports. For
    salesforce imports, the first and only element is the client
↳ credential
    id.
- remote_host_id : integer
```

Returns archived : string

The archival status of the requested object(s).

id : integer

The ID for the import.

running_as : dict:

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

schedule : dict:

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

syncs : list:

```
List of syncs.
- id : integer
- advanced_options : dict::
    - verify_table_row_counts : boolean
    - contact_lists : string
```

```

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

```

last_run : dict:

```

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

```

hidden : boolean

The hidden status of the object.

notifications : dict:

```

- failure_on : boolean
  If failure email notifications are on
- failure_email_addresses : list
  Addresses to notify by e-mail when the job fails.
- success_on : boolean
  If success email notifications are on
- stall_warning_minutes : integer
  Stall warning emails will be sent after this amount of minutes.
- success_email_subject : string

```

```
Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
    ↪successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
```

updated_at : string/date-time

user : dict:

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

source : dict:

```
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific,
    ↪imports. For
    ↪salesforce imports, the first and only element is the client,
    ↪credential
    ↪id.
- name : string
- remote_host_id : integer
```

is_outbound : boolean

next_run_at : string/time

The time of the next scheduled run.

destination : dict:

```
- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific,
    ↪imports. For
    ↪salesforce imports, the first and only element is the client,
    ↪credential
    ↪id.
- name : string
- remote_host_id : integer
```

parent_id : integer

Parent id to trigger this import from

sync_type : string

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

time_zone : string

The time zone of this import.

state : string

name : string

The name of the import.

created_at : string/date-time

put_archive (*id*, *status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **archived** : string

The archival status of the requested object(s).

id : integer

The ID for the import.

running_as : dict:

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

schedule : dict:

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

syncs : list:

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

last_run : dict:

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

hidden : boolean

The hidden status of the object.

notifications : dict:

```
- failure_on : boolean
  If failure email notifications are on
- failure_email_addresses : list
  Addresses to notify by e-mail when the job fails.
```

```

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

```

updated_at : string/date-time

user : dict:

```

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

```

source : dict:

```

- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific,
↳ imports. For
    salesforce imports, the first and only element is the client,
↳ credential
    id.
- name : string
- remote_host_id : integer

```

is_outbound : boolean

next_run_at : string/time

The time of the next scheduled run.

destination : dict:

```

- credential_id : integer
- additional_credentials : list
    Array that holds additional credentials used for specific,
↳ imports. For
    salesforce imports, the first and only element is the client,
↳ credential
    id.
- name : string
- remote_host_id : integer

```

parent_id : integer

Parent id to trigger this import from

sync_type : string

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

time_zone : string

The time zone of this import.

state : string

name : string

The name of the import.

created_at : string/date-time

put_projects (*id*, *project_id*)

Add a JobTypes::Import to a project

Parameters id : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

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

Set the permissions groups has on this object

Parameters id : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

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

Returns readers : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

put_shares_users (*id*, *permission_level*, *user_ids*)
Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

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

user_ids : list

An array of one or more user IDs

Returns **readers** : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

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

Update a sync

Parameters *id* : integer

The ID of the import to fetch.

sync_id : integer

The ID of the sync to fetch.

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.
```

advanced_options : dict, optional:

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

Returns `id` : integer

advanced_options : dict:

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

destination : dict:

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

source : dict:

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

Jobs

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

Methods

<code>delete_projects(id, project_id)</code>	Remove a Job from a project
<code>delete_shares_groups(id, group_id)</code>	Revoke the permissions a group has on this object
<code>delete_shares_users(id, user_id)</code>	Revoke the permissions a user has on this object
<code>get(id)</code>	Show basic job info
<code>list(**kwargs)</code>	List jobs
<code>list_children(id)</code>	Show nested tree of children that this job triggers
Continued on next page	

Table 4.8 – continued from previous page

<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, group_ids, ...)	Set the permissions groups has on this object
<i>put_shares_users</i> (id, permission_level, user_ids)	Set the permissions users have on this object

delete_projects (*id*, *project_id*)

Remove a Job from a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_shares_groups (*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters *id* : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users (*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters *id* : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get (*id*)

Show basic job info

Parameters *id* : integer

The ID for this job.

Returns *id* : integer**archived** : string

The archival status of the requested object(s).

created_at : string/date-time

last_run : dict:

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

state : string

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

hidden : boolean

The hidden status of the object.

type : string

updated_at : string/date-time

name : string

runs : list:

```
Information about the most recent runs of the job.
- 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.
- started_at : string/time
    The time that the run started.
```

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 **id** : integer

archived : string

The archival status of the requested object(s).

created_at : string/date-time

last_run : dict:

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

state : string

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

type : string

updated_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 **children** : list

id : integer

created_at : string/date-time

last_run : dict:

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

state : string
type : string
updated_at : string/date-time
name : string
runs : list:

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

list_parents (*id*)

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

Parameters **id** : integer

The ID for this job.

Returns **id** : integer

archived : string

The archival status of the requested object(s).

created_at : string/date-time

last_run : dict:

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

state : string

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

hidden : boolean

The hidden status of the object.

type : string

updated_at : string/date-time

name : string

runs : list:

```
Information about the most recent runs of the job.
- 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.
- started_at : string/time
    The time that the run started.
```

list_projects (*id*)

List the projects a Job belongs to

Parameters *id* : integer

The ID of the resource.

Returns *id* : integer

The ID for this project.

archived : string

The archival status of the requested object(s).

author : dict:

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

created_at : string/time

description : string

A description of the project

updated_at : string/time

name : string

The name of this project.

auto_share : boolean

users : list:

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

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

list_shares (*id*)

List users and groups permissioned on this object

Parameters *id* : integer

The ID of the object.

Returns *readers* : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

post_runs (*id*)

Run a job

Parameters *id* : integer

The ID for this job.

Returns None

Response code 204: success

post_trigger_email (*id*)

Generate and retrieve trigger email address

Parameters `id` : integer

The ID for this job.

Returns `trigger_email` : string

Email address which may be used to trigger this job to run.

put_projects (*id, project_id*)

Add a Job to a project

Parameters `id` : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_shares_groups (*id, group_ids, permission_level*)

Set the permissions groups has on this object

Parameters `id` : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

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

Returns `readers` : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

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

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

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

user_ids : list

An array of one or more user IDs

Returns **readers** : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

Models

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

Methods

<i>delete_builds</i> (id, build_id)	Cancel a build
<i>delete_projects</i> (id, project_id)	Remove a models 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)	Retrieve model configuration
<i>get_builds</i> (id, build_id)	Check status of a build
<i>list</i> (**kwargs)	List models
<i>list_builds</i> (id, **kwargs)	List builds for the given model
<i>list_projects</i> (id)	List the projects a models belongs to
<i>list_schedules</i> (id)	Show the model build schedule
<i>list_shares</i> (id)	List users and groups permissioned on this object
<i>list_types</i> ()	List all available model types
<i>patch</i> (id, **kwargs)	Update model configuration
<i>post</i> (**kwargs)	Create new configuration for a model
<i>post_builds</i> (id)	Start a build
<i>put_archive</i> (id, status)	Update the archive status of this object
<i>put_predictions</i> (id, table_name, primary_key, ...)	Add a table on which to apply the predictive model
<i>put_projects</i> (id, project_id)	Add a models to a project
<i>put_schedules</i> (id, schedule)	Schedule the model build
<i>put_shares_groups</i> (id, group_ids, ...)	Set the permissions groups has on this object
<i>put_shares_users</i> (id, permission_level, user_ids)	Set the permissions users have on this object

delete_builds (*id*, *build_id*)

Cancel a build

Parameters **id** : integer

The ID of the model.

build_id : integer

The ID of the build.

Returns None

Response code 202: success

delete_projects (*id*, *project_id*)

Remove a models from a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_shares_groups (*id, group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users (*id, user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get (*id*)

Retrieve model configuration

Parameters **id** : integer

The ID of the model.

Returns **excluded_columns** : list

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

id : integer

The ID of the model.

archived : string

The archival status of the requested object(s).

builds : list:

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

```
A key metric for binary, multinomial, and ordinal models. Nil_
↳ for other
  model types.
```

last_run : dict:

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

model_type_id : integer

The ID of the model's type.

updated_at : string/date-time

The time the model was updated.

number_of_folds : integer

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

time_zone : string

The time zone of this model.

running_as : dict:

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

notifications : dict:

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

```

    Addresses to notify by e-mail when the job completes.
    ↳ successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.

```

predictions : list:

```

The tables upon which the model will be applied.
- id : integer
    The ID of the model to which to apply the prediction.
- output_table : string
    The qualified name of the table to be created which will
    ↳ contain the
    model's predictions.
- schedule : dict::
    - scheduled : boolean
        If the object is scheduled
    - scheduled_minutes : list
        Minutes of the day it is scheduled on
    - scheduled_hours : list
        Hours of the day it is scheduled on
    - 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
- state : string
    The status of the prediction. One of: "succeeded", "failed",
    ↳ "queued",
    or "running", or "idle", if no build has been attempted.
- table_name : string
    The qualified name of the table on which to apply the
    ↳ predictive model.
- limiting_sql : string
    A SQL WHERE clause used to scope the rows to be predicted.
- primary_key : list
    The primary key or composite keys of the table being predicted.

```

current_build_exception : string

Exception message, if applicable, of the current model build.

description : string

A description of the model.

last_output_location : string

The output JSON for the last build.

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]}.

model_name : string

The name of the model.

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.

schedule : dict:

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

database_id : integer

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

dependent_variable_order : list

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

interaction_terms : boolean

Whether to search for interaction terms.

primary_key : string

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

hidden : boolean

The hidden status of the object.

dependent_variable : string

The dependent variable of the training dataset.

parent_id : integer

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

table_name : string

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

credential_id : integer

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

user : dict:

```

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

```

limiting_sql : string

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

created_at : string/date-time

The time the model was created.

get_builds (*id*, *build_id*)

Check status of a build

Parameters **id** : integer

The ID of the model.

build_id : integer

The ID of the build.

Returns **id** : integer

The ID of the model build.

error : string

The error, if any, returned by the build.

roc_auc : number/float

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

name : string

The name of the model build.

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.

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.

created_at : string

The time the model build was created.

description : string

A description of the model build.

state : string

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

transformation_metadata : string

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

output : string

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

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 **excluded_columns** : list

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

id : integer

The ID of the model.

archived : string

The archival status of the requested object(s).

builds : list:

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

last_run : dict:

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

description : string

A description of the model.

updated_at : string/date-time

The time the model was updated.

number_of_folds : integer

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

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.

model_type_id : integer

The ID of the model's type.

predictions : list:

The tables upon which the model will be applied.

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

current_build_exception : string

Exception message, if applicable, of the current model build.

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]}.

model_name : string

The name of the 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.

box_cox_transformation : boolean

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

schedule : dict:

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

```

database_id : integer

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

dependent_variable_order : list

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

interaction_terms : boolean

Whether to search for interaction terms.

primary_key : string

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

dependent_variable : string

The dependent variable of the training dataset.

parent_id : integer

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

table_name : string

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

time_zone : string

The time zone of this model.

user : dict:

```

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

```

limiting_sql : string

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

created_at : string/date-time

The time the model was created.

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 id : integer

The ID of the model build.

error : string

The error, if any, returned by the build.

roc_auc : number/float

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

name : string

The name of the model build.

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.

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.

created_at : string

The time the model build was created.

description : string

A description of the model build.

state : string

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

transformation_metadata : string

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

output : string

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

list_projects (*id*)

List the projects a models belongs to

Parameters **id** : integer

The ID of the resource.

Returns **id** : integer

The ID for this project.

archived : string

The archival status of the requested object(s).

author : dict:

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

created_at : string/time

description : string

A description of the project

updated_at : string/time

name : string

The name of this project.

auto_share : boolean

users : list:

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

list_schedules (*id*)

Show the model build schedule

Parameters **id** : integer

The ID of the model associated with this schedule.

Returns `id` : integer

The ID of the model associated with this schedule.

schedule : dict:

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

list_shares (*id*)

List users and groups permissioned on this object

Parameters `id` : integer

The ID of the object.

Returns `readers` : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

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

list_types()

List all available model types

Returns **dv_type** : string

The type of dependent variable predicted by the model.

id : integer

The ID of the model type.

int_allowed : boolean

Whether this model type supports searching for interaction terms.

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.

active_build_id : integer, optional

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

excluded_columns : list, optional

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

notifications : dict, optional:

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

box_cox_transformation : boolean, optional

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

schedule : dict, optional:

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

database_id : integer, optional

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

dependent_variable_order : list, optional

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

interaction_terms : boolean, optional

Whether to search for interaction terms.

model_type_id : integer, optional

The ID of the model's type.

primary_key : string, optional

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

number_of_folds : integer, optional

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

time_zone : string, optional

The time zone of this model.

table_name : string, optional

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

parent_id : integer, optional

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

description : string, optional

A description of the model.

credential_id : integer, optional

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

limiting_sql : string, optional

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

dependent_variable : string, optional

The dependent variable of the training dataset.

model_name : string, optional

The name of the model.

Returns None

Response code 204: success

post (***kwargs*)

Create new configuration for a model

Parameters **active_build_id** : integer, optional

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

excluded_columns : list, optional

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

notifications : dict, optional:

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

box_cox_transformation : boolean, optional

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

schedule : dict, optional:

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

database_id : integer, optional

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

dependent_variable_order : list, optional

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

interaction_terms : boolean, optional

Whether to search for interaction terms.

model_type_id : integer, optional

The ID of the model's type.

hidden : boolean, optional

The hidden status of the object.

primary_key : string, optional

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

number_of_folds : integer, optional

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

time_zone : string, optional

The time zone of this model.

table_name : string, optional

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

parent_id : integer, optional

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

description : string, optional

A description of the model.

credential_id : integer, optional

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

limiting_sql : string, optional

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

dependent_variable : string, optional

The dependent variable of the training dataset.

model_name : string, optional

The name of the model.

Returns excluded_columns : list

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

id : integer

The ID of the model.

archived : string

The archival status of the requested object(s).

builds : list:

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

last_run : dict:

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

model_type_id : integer

The ID of the model's type.

updated_at : string/date-time

The time the model was updated.

number_of_folds : integer

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

time_zone : string

The time zone of this model.

running_as : dict:

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

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

notifications : dict:

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

predictions : list:

```
The tables upon which the model will be applied.
- id : integer
    The ID of the model to which to apply the prediction.
- output_table : string
    The qualified name of the table to be created which will_
↳contain the
    model's predictions.
- schedule : dict::
    - scheduled : boolean
        If the object is scheduled
    - scheduled_minutes : list
        Minutes of the day it is scheduled on
    - scheduled_hours : list
        Hours of the day it is scheduled on
    - 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
- state : string
    The status of the prediction. One of: "succeeded", "failed",
↳"queued",
    or "running", or "idle", if no build has been attempted.
- table_name : string
    The qualified name of the table on which to apply the_
↳predictive model.
- limiting_sql : string
    A SQL WHERE clause used to scope the rows to be predicted.
- primary_key : list
```

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

current_build_exception : string

Exception message, if applicable, of the current model build.

description : string

A description of the model.

last_output_location : string

The output JSON for the last build.

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]}.

model_name : string

The name of the model.

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.

schedule : dict:

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

database_id : integer

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

dependent_variable_order : list

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

interaction_terms : boolean

Whether to search for interaction terms.

primary_key : string

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

hidden : boolean

The hidden status of the object.

dependent_variable : string

The dependent variable of the training dataset.

parent_id : integer

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

table_name : string

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

credential_id : integer

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

user : dict:

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

limiting_sql : string

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

created_at : string/date-time

The time the model was created.

post_builds (*id*)

Start a build

Parameters **id** : integer

The ID of the model.

Returns **id** : integer

The ID of the model build.

error : string

The error, if any, returned by the build.

roc_auc : number/float

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

name : string

The name of the model build.

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.

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.

created_at : string

The time the model build was created.

description : string

A description of the model build.

state : string

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

transformation_metadata : string

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

output : string

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

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 **excluded_columns** : list

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

id : integer

The ID of the model.

archived : string

The archival status of the requested object(s).

builds : list:

```
A list of trained models available for making predictions.
- root_mean_squared_error : number/float
  A key metric for continuous models. Nil for other model types.
- id : integer
  The ID of the model build.
- created_at : string
  The time the model build was created.
```

```
- description : string
    A description 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.
- roc_auc : number/float
    A key metric for binary, multinomial, and ordinal models. Nil,
↳ for other
    model types.
```

last_run : dict:

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

model_type_id : integer

The ID of the model's type.

updated_at : string/date-time

The time the model was updated.

number_of_folds : integer

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

time_zone : string

The time zone of this model.

running_as : dict:

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

notifications : dict:

```
- failure_on : boolean
    If failure email notifications are on
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
```

```

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

```

predictions : list:

```

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

```

current_build_exception : string

Exception message, if applicable, of the current model build.

description : string

A description of the model.

last_output_location : string

The output JSON for the last build.

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]}.

model_name : string

The name of the model.

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.

schedule : dict:

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

database_id : integer

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

dependent_variable_order : list

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

interaction_terms : boolean

Whether to search for interaction terms.

primary_key : string

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

hidden : boolean

The hidden status of the object.

dependent_variable : string

The dependent variable of the training dataset.

parent_id : integer

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

table_name : string

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

credential_id : integer

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

user : dict:

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

limiting_sql : string

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

created_at : string/date-time

The time the model was created.

put_predictions (*id*, *table_name*, *primary_key*, ***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.

table_name : string

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

primary_key : list

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

limiting_sql : string, optional

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

schedule : dict, optional:

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

output_table : string, optional

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

Returns id : integer

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

output_table : string

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

schedule : dict:

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

state : string

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

table_name : string

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

limiting_sql : string

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

primary_key : list

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

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

```

    Hours of the day it is scheduled on
- 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

```

Returns id : integer

The ID of the model associated with this schedule.

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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

```

put_shares_groups (*id, group_ids, permission_level*)

Set the permissions groups has on this object

Parameters id : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

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

Returns readers : dict:

```

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

```

total_group_shares : integer

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

owners : dict:

```

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

writers : dict:

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

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

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

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

user_ids : list

An array of one or more user IDs

Returns **readers** : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

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 *schedule* : dict:

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

id : integer

The ID of the prediction.

error : string

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

scored_table_id : integer

The ID of the source table for this prediction.

scored_table_name : string

The name of the source table for this prediction.

last_run : dict:

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

primary_key : list

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

model_id : integer

The ID of the model used for this prediction.

limiting_sql : string

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

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.

started_at : string/date-time

The start time of the last run of this prediction.

scored_tables : list:

```
An array of created prediction tables.
- schema : string
    The schema of table with created predictions.
- score_stats : list::
    An array of metrics on the created predictions.
    - score_name : string
```

```

    The name of the score.
    - avg_score : number/float
      The average score.
    - histogram : list
      The histogram of the distribution of scores.
    - max_score : number/float
      The maximum score.
    - min_score : number/float
      The minimum score.
  - id : integer
    The ID of the table with created predictions.
  - name : string
    The name of table with created predictions.
  - created_at : string/date-time
    The time when the table with created predictions was created.

```

output_table_name : string

The name of the output table for this prediction.

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 *score_stats* : list:

```

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

```

id : integer

The ID of the prediction run.

exception : string

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

prediction_id : integer

The ID of the prediction.

state : string

The state of the prediction run.

name : string

The name of table created by this predictions run.

created_at : string/date-time

The time when the table with created predictions was created.

list (***kwargs*)

List predictions

Parameters **model_id** : integer, optional

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

Returns **model_id** : integer

The ID of the model used for this prediction.

id : integer

The ID of the prediction.

finished_at : string/date-time

The end time of the last run of this prediction.

scored_table_id : integer

The ID of the source table for this prediction.

last_run : dict:

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

scored_table_name : string

The name of the source table for this prediction.

state : string

The state of the last run of this prediction.

output_table_name : string

The name of the output table for this prediction.

error : string

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

started_at : string/date-time

The start time of the last run of this prediction.

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 **score_stats** : list:

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

id : integer

The ID of the prediction run.

exception : string

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

prediction_id : integer

The ID of the prediction.

state : string

The state of the prediction run.

name : string

The name of table created by this predictions run.

created_at : string/date-time

The time when the table with created predictions was created.

list_schedules (*id*)

Show the prediction schedule

Parameters **id** : integer

ID of the prediction associated with this schedule.

Returns **id** : integer

ID of the prediction associated with this schedule.

schedule : dict:

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

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.

limiting_sql : string, optional

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

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.

Returns **schedule** : dict:

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

id : integer

The ID of the prediction.

error : string

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

scored_table_id : integer

The ID of the source table for this prediction.

scored_table_name : string

The name of the source table for this prediction.

last_run : dict:

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

primary_key : list

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

model_id : integer

The ID of the model used for this prediction.

limiting_sql : string

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

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.

started_at : string/date-time

The start time of the last run of this prediction.

scored_tables : list:

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

output_table_name : string

The name of the output table for this prediction.

post_runs (*id*)

Start a run

Parameters *id* : integer

The ID of the prediction.

Returns *score_stats* : list:

An array of metrics on the created predictions.

- *score_name* : string
The name of the score.
- *avg_score* : number/float
The average score.
- *histogram* : list
The histogram of the distribution of scores.
- *max_score* : number/float
The maximum score.
- *min_score* : number/float
The minimum score.

id : integer

The ID of the prediction run.

exception : string

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

prediction_id : integer

The ID of the prediction.

state : string

The state of the prediction run.

name : string

The name of table created by this predictions run.

created_at : string/date-time

The time when the table with created predictions was created.

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_minutes* : list
Minutes of the day it *is* scheduled on
- *scheduled_hours* : list
Hours of the day it *is* scheduled on

```

- 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

```

score_on_model_build : boolean, optional

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

Returns id : integer

ID of the prediction associated with this schedule.

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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

```

score_on_model_build : boolean

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

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(database, preview_rows, 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 *last_run_id* : integer

The ID of the last run.

id : integer

The query ID.

script_id : integer

The ID of the script associated with this query.

author : dict:

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

exception : string

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

hidden : boolean

The hidden status of the object.

name : string

The name of the query.

updated_at : string/time

database : integer

The database ID.

report_id : integer

The ID of the report associated with this query.

created_at : string/time

state : string

The state of the last run.

result_columns : list

A preview of columns returned by the query.

finished_at : string/date-time

The end time of the last run.

result_rows : list

A preview of rows returned by the query.

started_at : string/date-time

The start time of the last run.

credential : integer

The credential ID.

sql : string

The SQL to execute.

get_runs (*id*, *run_id*)

Check status of a run

Parameters **id** : integer

The ID of the query.

run_id : integer

The ID of the run.

Returns **id** : integer

The ID of the run.

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.

started_at : string/time

The time the last run started at.

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 **last_run_id** : integer

The ID of the last run.

id : integer

The query ID.

script_id : integer

The ID of the script associated with this query.

exception : string

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

updated_at : string/time

database : integer

The database ID.

preview_rows : integer

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

report_id : integer

The ID of the report associated with this query.

created_at : string/time

state : string

The state of the last run.

result_columns : list

A preview of columns returned by the query.

finished_at : string/date-time

The end time of the last run.

result_rows : list

A preview of rows returned by the query.

started_at : string/date-time

The start time of the last run.

credential : integer

The credential ID.

sql : string

The SQL to execute.

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 **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.

started_at : string/time

The time the last run started at.

post (*database*, *preview_rows*, *sql*, ***kwargs*)

Execute a query

Parameters **database** : integer

The database ID.

preview_rows : integer

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

sql : string

The SQL to execute.

filename_prefix : string, optional

The output filename prefix.

compression : string, optional

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

unquoted : boolean, optional

If true, will not quote fields.

hidden : boolean, optional

The hidden status of the object.

column_delimiter : string, optional

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

credential : integer, optional

The credential ID.

include_header : boolean, optional

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

interactive : boolean, optional

Deprecated and not used.

Returns **last_run_id** : integer

The ID of the last run.

id : integer

The query ID.

script_id : integer

The ID of the script associated with this query.

compression : string

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

updated_at : string/time

interactive : boolean

Deprecated and not used.

database : integer

The database ID.

unquoted : boolean

If true, will not quote fields.

filename_prefix : string

The output filename prefix.

state : string

The state of the last run.

column_delimiter : string

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

finished_at : string/date-time

The end time of the last run.

result_rows : list

A preview of rows returned by the query.

started_at : string/date-time

The start time of the last run.

credential : integer

The credential ID.

preview_rows : integer

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

exception : string

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

hidden : boolean

The hidden status of the object.

result_columns : list

A preview of columns returned by the query.

created_at : string/time

report_id : integer

The ID of the report associated with this query.

include_header : boolean

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

sql : string

The SQL to execute.

post_runs (*id*)

Start a run

Parameters *id* : integer

The ID of the query.

Returns *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.

started_at : string/time

The time the last run started at.

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 *last_run_id* : integer

The ID of the last run.

id : integer

The query ID.

script_id : integer

The ID of the script associated with this query.

author : dict:

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

```
- online : boolean
    Whether this user is online.
```

exception : string

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

hidden : boolean

The hidden status of the object.

name : string

The name of the query.

updated_at : string/time

database : integer

The database ID.

report_id : integer

The ID of the report associated with this query.

created_at : string/time

state : string

The state of the last run.

result_columns : list

A preview of columns returned by the query.

finished_at : string/date-time

The end time of the last run.

result_rows : list

A preview of rows returned by the query.

started_at : string/date-time

The start time of the last run.

credential : integer

The credential ID.

sql : string

The SQL to execute.

Reports

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

Methods

delete_grants(id)

Revoke permission for this report to perform Civis platform API operations on

Continued on next page

Table 4.12 – continued from previous page

<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>patch</i> (id, **kwargs)	Update a report
<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>put_archive</i> (id, status)	Update the archive status of this object
<i>put_projects</i> (id, project_id)	Add a Report to a project
<i>put_shares_groups</i> (id, group_ids, ...)	Set the permissions groups has on this object
<i>put_shares_users</i> (id, permission_level, user_ids)	Set the permissions users have on this object

delete_grants (*id*)

Revoke permission for this report to perform Civis platform API operations on your behalf

Parameters *id* : integer

The ID of this report.

Returns None

Response code 204: success

delete_projects (*id*, *project_id*)

Remove a Report from a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_shares_groups (*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters *id* : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users (*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters *id* : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get (*id*)

Show a single report

Parameters **id** : integer

The ID of this report.

Returns **id** : integer

The ID of this report.

config : string

Any configuration metadata for this report.

last_run : dict:

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

provide_api_key : boolean

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

updated_at : string/time

script : dict:

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

archived : string

The archival status of the requested object(s).

api_key : string

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

state : string

The status of the report's last run.

auth_thumbnail_url : string

URL for a thumbnail of the report.

template_id : integer

The ID of the template used for this report.

valid_output_file : boolean

Whether the script that backs the report currently has a valid output file.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

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

api_key_id : integer

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

hidden : boolean

The hidden status of the object.

projects : list:

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

auth_code_url : string

app_state : dict

Any application state blob for this report.

created_at : string/time

user : dict:

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

auth_data_url : string

name : string

The name of the report.

tableau_id : integer

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 **template_id** : integer

The ID of the template used for this report.

id : integer

The ID of this report.

auth_thumbnail_url : string

URL for a thumbnail of the report.

viz_updated_at : string/time

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

last_run : dict:

```
- 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.
```

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

updated_at : string/time

projects : list:

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

script : dict:

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

archived : string

The archival status of the requested object(s).

created_at : string/time

user : dict:

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

state : string

The status of the report's last run.

finished_at : string/time

The time that the report's last run finished.

name : string

The name of the report.

tableau_id : integer

list_projects (*id*)

List the projects a Report belongs to

Parameters **id** : integer

The ID of the resource.

Returns **id** : integer

The ID for this project.

archived : string

The archival status of the requested object(s).

author : dict:

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

created_at : string/time

description : string

A description of the project

updated_at : string/time

name : string

The name of this project.

auto_share : boolean

users : list:

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

list_shares (*id*)

List users and groups permissioned on this object

Parameters *id* : integer

The ID of the object.

Returns *readers* : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

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

Update a report

Parameters **id** : integer

The ID of the report to modify.

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.

code_body : string, optional

The code for the report visualization.

script_id : integer, optional

The ID of the script used to create this report.

config : string, optional

app_state : dict, optional

The application state blob for this report.

provide_api_key : boolean, optional

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

name : string, optional

The name of the report.

Returns **id** : integer

The ID of this report.

config : string

Any configuration metadata for this report.

last_run : dict:

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

provide_api_key : boolean

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

updated_at : string/time

script : dict:

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

archived : string

The archival status of the requested object(s).

api_key : string

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

state : string

The status of the report's last run.

auth_thumbnail_url : string

URL for a thumbnail of the report.

template_id : integer

The ID of the template used for this report.

valid_output_file : boolean

Whether the script that backs the report currently has a valid output file.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

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

api_key_id : integer

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

hidden : boolean

The hidden status of the object.

projects : list:

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

auth_code_url : string

app_state : dict

Any application state blob for this report.

created_at : string/time

user : dict:

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

auth_data_url : string

name : string

The name of the report.

tableau_id : integer

post (**kwargs)

Create a report

Parameters **template_id** : integer, optional

The ID of the template used for this report.

code_body : string, optional

The code for the report visualization.

script_id : integer, optional

The ID of the script used to create this report.

app_state : dict, optional

Any application state blob for this report.

provide_api_key : boolean, optional

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

hidden : boolean, optional

The hidden status of the object.

name : string, optional

The name of the report.

Returns id : integer

The ID of this report.

config : string

Any configuration metadata for this report.

last_run : dict:

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

provide_api_key : boolean

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

updated_at : string/time

script : dict:

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

archived : string

The archival status of the requested object(s).

api_key : string

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

state : string

The status of the report's last run.

auth_thumbnail_url : string

URL for a thumbnail of the report.

template_id : integer

The ID of the template used for this report.

valid_output_file : boolean

Whether the script that backs the report currently has a valid output file.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

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

api_key_id : integer

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

hidden : boolean

The hidden status of the object.

projects : list:

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

auth_code_url : string

app_state : dict

Any application state blob for this report.

created_at : string/time

user : dict:

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

auth_data_url : string

name : string

The name of the report.

tableau_id : integer

post_grants (*id*)

Grant this report the ability to perform Civis platform API operations on your behalf

Parameters id : integer

The ID of this report.

Returns id : integer

The ID of this report.

config : string

Any configuration metadata for this report.

last_run : dict:


```

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

```

provide_api_key : boolean

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

updated_at : string/time

script : dict:

```

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

```

archived : string

The archival status of the requested object(s).

api_key : string

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

state : string

The status of the report's last run.

auth_thumbnail_url : string

URL for a thumbnail of the report.

template_id : integer

The ID of the template used for this report.

valid_output_file : boolean

Whether the script that backs the report currently has a valid output file.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

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

api_key_id : integer

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

hidden : boolean

The hidden status of the object.

projects : list:

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

auth_code_url : string

app_state : dict

Any application state blob for this report.

created_at : string/time

user : dict:

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

auth_data_url : string

name : string

The name of the report.

tableau_id : integer

put_archive (*id*, *status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **id** : integer

The ID of this report.

config : string

Any configuration metadata for this report.

last_run : dict:

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

```

provide_api_key : boolean

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

updated_at : string/time

script : dict:

```

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

```

archived : string

The archival status of the requested object(s).

api_key : string

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

state : string

The status of the report's last run.

auth_thumbnail_url : string

URL for a thumbnail of the report.

template_id : integer

The ID of the template used for this report.

valid_output_file : boolean

Whether the script that backs the report currently has a valid output file.

finished_at : string/time

The time that the report's last run finished.

viz_updated_at : string/time

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

api_key_id : integer

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

hidden : boolean

The hidden status of the object.

projects : list:

```

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

```

auth_code_url : string

app_state : dict

Any application state blob for this report.

created_at : string/time

user : dict:

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

auth_data_url : string

name : string

The name of the report.

tableau_id : integer

put_projects (*id*, *project_id*)

Add a Report to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

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

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

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

Returns **readers** : dict:

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

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

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

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

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

user_ids : list

An array of one or more user IDs

Returns **readers** : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

Scripts

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

Methods

<code>delete(id)</code>	Delete a script
<code>delete_containers(id)</code>	Delete a container
<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(id)</code>	Delete a CustomScript
<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(id)</code>	Delete a JavaScript Script
<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(id)</code>	Delete a Python Script
<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

Continued on next page

Table 4.13 – continued from previous page

<i>delete_python3_shares_groups</i> (id, group_id)	Revoke the permissions a group has on this object
<i>delete_python3_shares_users</i> (id, user_id)	Revoke the permissions a user has on this object
<i>delete_r</i> (id)	Delete an R Script
<i>delete_r_projects</i> (id, project_id)	Remove a r docker from a project
<i>delete_r_runs</i> (id, run_id)	Cancel a run
<i>delete_r_shares_groups</i> (id, group_id)	Revoke the permissions a group has on this object
<i>delete_r_shares_users</i> (id, user_id)	Revoke the permissions a user has on this object
<i>delete_sql</i> (id)	Delete a SQL script
<i>delete_sql_projects</i> (id, project_id)	Remove a scripts from a project
<i>delete_sql_runs</i> (id, run_id)	Cancel a run
<i>delete_sql_shares_groups</i> (id, group_id)	Revoke the permissions a group has on this object
<i>delete_sql_shares_users</i> (id, user_id)	Revoke the permissions a user has on this object
<i>get</i> (id)	Get details about a script
<i>get_containers</i> (id)	View a container
<i>get_containers_runs</i> (id, run_id)	Check status of a run
<i>get_custom</i> (id)	Get a CustomScript
<i>get_custom_runs</i> (id, run_id)	Check status of a run
<i>get_javascript</i> (id)	Get a JavaScript Script
<i>get_javascript_runs</i> (id, run_id)	Check status of a run
<i>get_python3</i> (id)	Get a Python Script
<i>get_python3_runs</i> (id, run_id)	Check status of a run
<i>get_r</i> (id)	Get an R Script
<i>get_r_runs</i> (id, run_id)	Check status of a run
<i>get_sql</i> (id)	Get a SQL script
<i>get_sql_runs</i> (id, run_id)	Check status of a run
<i>list</i> (**kwargs)	List scripts
<i>list_containers_projects</i> (id)	List the projects a container docker belongs to
<i>list_containers_runs</i> (id, **kwargs)	List runs for the given container
<i>list_containers_runs_logs</i> (id, run_id, **kwargs)	Get the logs for a run
<i>list_containers_shares</i> (id)	List users and groups permissioned on this object
<i>list_custom</i> (**kwargs)	List Custom Scripts
<i>list_custom_projects</i> (id)	List the projects a Job belongs to
<i>list_custom_runs</i> (id, **kwargs)	List runs for the given custom
<i>list_custom_runs_logs</i> (id, run_id, **kwargs)	Get the logs for a run
<i>list_custom_shares</i> (id)	List users and groups permissioned on this object
<i>list_history</i> (id)	Get the run history and outputs of this script
<i>list_javascript_projects</i> (id)	List the projects a scripted sql belongs to
<i>list_javascript_runs</i> (id, **kwargs)	List runs for the given javascript
<i>list_javascript_runs_logs</i> (id, run_id, **kwargs)	Get the logs for a run
<i>list_javascript_shares</i> (id)	List users and groups permissioned on this object
<i>list_python3_projects</i> (id)	List the projects a python docker belongs to
<i>list_python3_runs</i> (id, **kwargs)	List runs for the given python
<i>list_python3_runs_logs</i> (id, run_id, **kwargs)	Get the logs for a run
<i>list_python3_shares</i> (id)	List users and groups permissioned on this object
<i>list_r_projects</i> (id)	List the projects a r docker belongs to
<i>list_r_runs</i> (id, **kwargs)	List runs for the given r
<i>list_r_runs_logs</i> (id, run_id, **kwargs)	Get the logs for a run

Continued on next page

Table 4.13 – continued from previous page

<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_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_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(credential_id, name, remote_host_id, ...)</i>	Create a script
<i>post_cancel(id)</i>	Cancel a run
<i>post_containers(docker_command, ...)</i>	Create a container
<i>post_containers_runs(id)</i>	Start a run
<i>post_custom(from_template_id, **kwargs)</i>	Create a CustomScript
<i>post_custom_runs(id)</i>	Start a run
<i>post_javascript(source, credential_id, name, ...)</i>	Create a JavaScript Script
<i>post_javascript_runs(id)</i>	Start a run
<i>post_python3(source, name, **kwargs)</i>	Create a Python Script
<i>post_python3_runs(id)</i>	Start a run
<i>post_r(source, name, **kwargs)</i>	Create an R Script
<i>post_r_runs(id)</i>	Start a run
<i>post_run(id)</i>	Run a script
<i>post_sql(credential_id, name, ...)</i>	Create a SQL script
<i>post_sql_runs(id)</i>	Start a run
<i>put_containers(id, docker_command, ...)</i>	Edit a container
<i>put_containers_archive(id, status)</i>	Update the archive status of this object
<i>put_containers_projects(id, project_id)</i>	Add a container docker to a project
<i>put_containers_shares_groups(id, group_ids, ...)</i>	Set the permissions groups has on this object
<i>put_containers_shares_users(id, ...)</i>	Set the permissions users have on this object
<i>put_custom(id, **kwargs)</i>	Replace all attributes of this CustomScript
<i>put_custom_archive(id, status)</i>	Update the archive status of this object
<i>put_custom_projects(id, project_id)</i>	Add a Job to a project
<i>put_custom_shares_groups(id, group_ids, ...)</i>	Set the permissions groups has on this object
<i>put_custom_shares_users(id, ...)</i>	Set the permissions users have on this object
<i>put_javascript(id, source, credential_id, ...)</i>	Replace all attributes of this JavaScript Script
<i>put_javascript_archive(id, status)</i>	Update the archive status of this object
<i>put_javascript_projects(id, project_id)</i>	Add a scripted sql to a project
<i>put_javascript_shares_groups(id, group_ids, ...)</i>	Set the permissions groups has on this object
<i>put_javascript_shares_users(id, ...)</i>	Set the permissions users have on this object
<i>put_python3(id, source, name, **kwargs)</i>	Replace all attributes of this Python Script
<i>put_python3_archive(id, status)</i>	Update the archive status of this object
<i>put_python3_projects(id, project_id)</i>	Add a python docker to a project
<i>put_python3_shares_groups(id, group_ids, ...)</i>	Set the permissions groups has on this object
<i>put_python3_shares_users(id, ...)</i>	Set the permissions users have on this object

Continued on next page

Table 4.13 – continued from previous page

<i>put_r</i> (id, source, name, **kwargs)	Replace all attributes of this R Script
<i>put_r_archive</i> (id, status)	Update the archive status of this object
<i>put_r_projects</i> (id, project_id)	Add a r docker to a project
<i>put_r_shares_groups</i> (id, group_ids, ...)	Set the permissions groups has on this object
<i>put_r_shares_users</i> (id, permission_level, ...)	Set the permissions users have on this object
<i>put_sql</i> (id, credential_id, name, ...)	Replace all attributes of this SQL script
<i>put_sql_archive</i> (id, status)	Update the archive status of this object
<i>put_sql_projects</i> (id, project_id)	Add a scripts to a project
<i>put_sql_shares_groups</i> (id, group_ids, ...)	Set the permissions groups has on this object
<i>put_sql_shares_users</i> (id, permission_level, ...)	Set the permissions users have on this object

delete (*id*)

Delete a script

Parameters *id* : integer

The ID for the script.

Returns None

Response code 204: success

delete_containers (*id*)

Delete a container

Parameters *id* : integer

The ID for the script.

Returns None

Response code 204: success

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 (*id*)

Delete a CustomScript

Parameters **id** : integer

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 (*id*)

Delete a JavaScript Script

Parameters **id** : integer

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 (*id*)

Delete a Python Script

Parameters **id** : integer

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 (*id*)

Delete an R Script

Parameters **id** : integer

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 (*id*)

Delete a SQL script

Parameters **id** : integer

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 *author* : dict:

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

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

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

params : list:

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

type : string

The type of script.

updated_at : string/time

The time this script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

template_script_id : integer

The ID of the template script, if any.

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

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

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

running_as : dict:

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

schedule : dict:

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

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

projects : list:

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

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

sql : string

The raw SQL query for the script.

get_containers (*id*)

View a container

Parameters **id** : integer

The ID for the script.

Returns **author** : dict:

```

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

```

id : integer

The ID for the script.

remote_host_credential_id : integer

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

user_context : string

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

docker_command : string

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

params : list:

```

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

```

type : string

The type of the script (e.g Container)

last_run : dict:

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```
- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
↪ core has
    1024 shares.
```

archived : string

The archival status of the requested object(s).

notifications : dict:

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

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
```

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

repo_http_uri : string

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

running_as : dict:

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

schedule : dict:

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

is_template : boolean

Whether others scripts use this one as a template.

repo_ref : string

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

template_script_name : string

The name of the template script.

docker_image_name : string

The name of the docker image to pull from DockerHub.

projects : list:

A `list` of projects containing the script.

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

template_dependents_count : integer

How many other scripts use this one as a template.

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

git_credential_id : integer

The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_image_tag : string

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

name : string

The name of the container.

from_template_id : integer

The ID of the template script.

get_containers_runs (*id*, *run_id*)

Check status of a run

Parameters **id** : integer

The ID of the container.

run_id : integer

The ID of the run.

Returns **id** : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

container_id : integer

The ID of the container.

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.

started_at : string/time

The time the last run started at.

get_custom (*id*)

Get a CustomScript

Parameters *id* : integer

Returns *id* : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

last_run : dict:

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

params : list:

```
A definition of the parameters this script accepts in the arguments_
↳field.
- 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.
- description : string
    A short sentence or fragment describing this parameter to the_
↳end user.
- required : boolean
    Whether this param is required.
```

```
- type : string
    The type of parameter. Valid options: string, integer, float, ↵
    ↵bool,
    database, credential_aws, credential_redshift, or credential_
    ↵custom
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
```

type : string

The type of the script (e.g Custom)

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

notifications : dict:

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

schedule : dict:

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

finished_at : string/time

The time that the script's last run finished.

code_preview : string

The code that this script will run with arguments inserted.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:

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

author : dict:

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

template_script_name : string

The name of the template script.

projects : list:

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

credential_id : integer

The credential that this script will use.

name : string

The name of the script.

from_template_id : integer

The ID of the template script.

state : string

The status of the script's last run.

get_custom_runs (*id*, *run_id*)

Check status of a run

Parameters **id** : integer

The ID of the custom.

run_id : integer

The ID of the run.

Returns **id** : integer

The ID of the run.

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.

started_at : string/time

The time the last run started at.

custom_id : integer

The ID of the custom.

get_javascript (*id*)

Get a JavaScript Script

Parameters **id** : integer

Returns **author** : dict:

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

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

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

params : list:

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

type : string

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

credential_id : integer

The credential that this script will use.

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

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

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:

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

schedule : dict:

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

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

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

source : string

The body/text of the script.

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

get_javascript_runs (*id*, *run_id*)

Check status of a run

Parameters *id* : integer

The ID of the javascript.

run_id : integer

The ID of the run.

Returns *id* : integer

The ID of the run.

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.

javascript_id : integer

The ID of the javascript.

started_at : string/time

The time the last run started at.

get_python3 (*id*)

Get a Python Script

Parameters *id* : integer

Returns *author* : dict:

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

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

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

params : list:

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

type : string

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```
- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
↪core has
    1024 shares. Must be at least 2 shares.
```

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

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

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

running_as : dict:

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



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

schedule : dict:

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

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

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

source : string

The body/text of the script.

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

get_python3_runs (*id*, *run_id*)

Check status of a run

Parameters **id** : integer

The ID of the python.

run_id : integer

The ID of the run.

Returns **id** : integer

The ID of the run.

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.

started_at : string/time

The time the last run started at.

get_r (*id*)

Get an R Script

Parameters **id** : integer

Returns **author** : dict:

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

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

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

params : list:

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

type : string

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```
- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has 1024 shares. Must be at least 2 shares.
```

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

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

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

running_as : dict:

```

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

```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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

```

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

```

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

```

source : string

The body/text of the script.

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

get_r_runs (*id*, *run_id*)

Check status of a run

Parameters **id** : integer

The ID of the r.

run_id : integer

The ID of the run.

Returns **id** : integer

The ID of the run.

error : string

The error, if any, returned by the run.

is_cancel_requested : boolean

True if run cancel requested, else false.

r_id : integer

The ID of the r.

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.

started_at : string/time

The time the last run started at.

get_sql (*id*)

Get a SQL script

Parameters **id** : integer

Returns **author** : dict:

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

```
- online : boolean
    Whether this user is online.
```

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

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

params : list:

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

type : string

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

credential_id : integer

The credential that this script will use.

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

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

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

code_preview : string

The code that this script will run with arguments inserted.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:


```

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

```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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

```

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

projects : list:

```

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

```

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

sql : string

The raw SQL query for the script.

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 **sql_id** : integer

The ID of this sql.

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.

state : string

The state of this run.

finished_at : string/time

The time that this run finished.

output : list:

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

started_at : string/time

The time the last run started.

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.

status : string, optional

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

archived : string, optional

The archival status of the requested object(s).

limit : integer, optional

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

page_num : integer, optional

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

order : string, optional

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

order_dir : string, optional

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

iterator : bool, optional

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

Returns **id** : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

author : dict:

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

last_run : dict:

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

type : string

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

updated_at : string/time

The time the script was last updated.

projects : list:

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

from_template_id : integer

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

template_script_id : integer

The ID of the template script, if any.

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

links : dict:

```
- details : string
    The details link to get more information about the script.
- runs : string
    The runs link to get the run information list for this script.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

name : string

The name of the script.

is_template : boolean

Whether others scripts use this one as a template.

list_containers_projects (*id*)

List the projects a container docker belongs to

Parameters *id* : integer

The ID of the resource.

Returns *id* : integer

The ID for this project.

archived : string

The archival status of the requested object(s).

author : dict:

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

created_at : string/time

description : string

A description of the project

updated_at : string/time

name : string

The name of this project.

auto_share : boolean

users : list:

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

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 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.

container_id : integer

The ID of the container.

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.

started_at : string/time

The time the last run started at.

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 message : string

The log message.

id : integer

The ID of the log.

level : string

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

created_at : string/date-time

The time the log was created.

list_containers_shares (*id*)

List users and groups permissioned on this object

Parameters id : integer

The ID of the object.

Returns readers : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

list_custom (**kwargs)

List Custom Scripts

Parameters **from_template_id** : integer, optional

The template script that this app uses.

archived : string, optional

The archival status of the requested object(s).

limit : integer, optional

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

page_num : integer, optional

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

order : string, optional

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

order_dir : string, optional

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

iterator : bool, optional

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

Returns **id** : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

author : dict:

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

last_run : dict:

```
- 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.
```



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

type : string

The type of the script (e.g Custom)

updated_at : string/time

The time the script was last updated.

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.

parent_id : integer

The ID of the parent job that will trigger this script

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

name : string

The name of the script.

from_template_id : integer

The ID of the template script.

list_custom_projects (*id*)

List the projects a Job belongs to

Parameters **id** : integer

The ID of the resource.

Returns **id** : integer

The ID for this project.

archived : string

The archival status of the requested object(s).

author : dict:

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

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

created_at : string/time

description : string

A description of the project

updated_at : string/time

name : string

The name of this project.

auto_share : boolean

users : list:

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

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 *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.

started_at : string/time

The time the last run started at.

custom_id : integer

The ID of the custom.

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 **message** : string

The log message.

id : integer

The ID of the log.

level : string

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

created_at : string/date-time

The time the log was created.

list_custom_shares (*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns `readers` : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

list_history (*id*)

Get the run history and outputs of this script

Parameters `id` : integer

The ID for the script.

Returns `sql_id` : integer

The ID of this sql.

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.

state : string

The state of this run.

finished_at : string/time

The time that this run finished.

output : list:

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

list_javascript_projects (*id*)

List the projects a scripted sql belongs to

Parameters **id** : integer

The ID of the resource.

Returns **id** : integer

The ID for this project.

archived : string

The archival status of the requested object(s).

author : dict:

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

created_at : string/time

description : string

A description of the project

updated_at : string/time

name : string

The name of this project.

auto_share : boolean

users : list:

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

```
- online : boolean
    Whether this user is online.
```

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 *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.

javascript_id : integer

The ID of the javascript.

started_at : string/time

The time the last run started at.

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 message : string

The log message.

id : integer

The ID of the log.

level : string

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

created_at : string/date-time

The time the log was created.

list_javascript_shares (*id*)

List users and groups permissioned on this object

Parameters id : integer

The ID of the object.

Returns readers : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

list_python3_projects (*id*)

List the projects a python docker belongs to

Parameters *id* : integer

The ID of the resource.

Returns *id* : integer

The ID for this project.

archived : string

The archival status of the requested object(s).

author : dict:

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

created_at : string/time

description : string

A description of the project

updated_at : string/time

name : string

The name of this project.

auto_share : boolean

users : list:

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


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 *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.

started_at : string/time

The time the last run started at.

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 message : string

The log message.

id : integer

The ID of the log.

level : string

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

created_at : string/date-time

The time the log was created.

list_python3_shares (*id*)

List users and groups permissioned on this object

Parameters id : integer

The ID of the object.

Returns readers : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

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

list_r_projects (*id*)

List the projects a r docker belongs to

Parameters *id* : integer

The ID of the resource.

Returns *id* : integer

The ID for this project.

archived : string

The archival status of the requested object(s).

author : dict:

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

created_at : string/time

description : string

A description of the project

updated_at : string/time

name : string

The name of this project.

auto_share : boolean

users : list:

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

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 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.

r_id : integer

The ID of the r.

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.

started_at : string/time

The time the last run started at.

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 message : string

The log message.

id : integer

The ID of the log.

level : string

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

created_at : string/date-time

The time the log was created.

list_r_shares (*id*)

List users and groups permissioned on this object

Parameters id : integer

The ID of the object.

Returns readers : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

list_sql_projects (*id*)

List the projects a scripts belongs to

Parameters *id* : integer

The ID of the resource.

Returns *id* : integer

The ID for this project.

archived : string

The archival status of the requested object(s).

author : dict:

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

created_at : string/time

description : string

A description of the project

updated_at : string/time

name : string

The name of this project.

auto_share : boolean

users : list:

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

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 sql_id : integer

The ID of this sql.

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.

state : string

The state of this run.

finished_at : string/time

The time that this run finished.

output : list:

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

started_at : string/time

The time the last run started.

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 message : string

The log message.

id : integer

The ID of the log.

level : string

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

created_at : string/date-time

The time the log was created.

list_sql_shares (*id*)

List users and groups permissioned on this object

Parameters id : integer

The ID of the object.

Returns readers : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:


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

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.

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.

arguments : dict, optional

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

schedule : dict, optional:

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

```
- failure_on : boolean
  If failure email notifications are on
- failure_email_addresses : list
  Addresses to notify by e-mail when the job fails.
- success_on : boolean
  If success email notifications are on
- stall_warning_minutes : integer
  Stall warning emails will be sent after this amount of minutes.
- success_email_subject : string
  Custom subject line for success e-mail.
- urls : list
```

```
URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
    ↪ successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
```

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.
- 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.
- description : string
    A short sentence or fragment describing this parameter to the_
    ↪ end user.
- required : boolean
    Whether this param is required.
- type : string
    The type of parameter. Valid options: string, integer, float,_
    ↪ bool,
    database, credential_aws, credential_redshift, or credential_
    ↪ custom
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
```

name : string, optional

The name of the script.

sql : string, optional

The raw SQL query for the script.

Returns author : dict:

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

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

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

params : list:

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

type : string

The type of script.

updated_at : string/time

The time this script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

template_script_id : integer

The ID of the template script, if any.

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

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

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

running_as : dict:

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

schedule : dict:

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

```

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

```

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

projects : list:

```

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

```

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

sql : string

The raw SQL query for the script.

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

Update a container

Parameters *id* : integer

The ID for the script.

remote_host_credential_id : integer, optional

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

user_context : string, optional

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

docker_command : string, optional

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

repo_ref : string, optional

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

params : list, optional:

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

git_credential_id : integer, optional

The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you’ve submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_image_name : string, optional

The name of the docker image to pull from DockerHub.

required_resources : dict, optional:

```

- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
↪ core has
    1024 shares.

```

target_project_id : integer, optional

Target project to which script outputs will be added.

arguments : dict, optional

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

docker_image_tag : string, optional

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

```

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

```

schedule : dict, optional:

```

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

name : string, optional

The name of the container.

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.

Returns **author** : dict:

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

id : integer

The ID for the script.

remote_host_credential_id : integer

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

user_context : string

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

docker_command : string

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

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- 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.
- description : string
  A short sentence or fragment describing this parameter to the_
  ↪end user.
- required : boolean
  Whether this param is required.
- type : string
```



```

    The type of parameter. Valid options: string, integer, float, ↵
    ↵bool,
    database, credential_aws, credential_redshift, or credential_
    ↵custom
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.

```

type : string

The type of the script (e.g Container)

last_run : dict:

```

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

```

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```

- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each ↵
    ↵core has
    1024 shares.

```

archived : string

The archival status of the requested object(s).

notifications : dict:

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

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

repo_http_uri : string

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

running_as : dict:

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

schedule : dict:

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

```

is_template : boolean

Whether others scripts use this one as a template.

repo_ref : string

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

template_script_name : string

The name of the template script.

docker_image_name : string

The name of the docker image to pull from DockerHub.

projects : list:

```

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

```

template_dependents_count : integer

How many other scripts use this one as a template.

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

git_credential_id : integer

The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_image_tag : string

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

name : string

The name of the container.

from_template_id : integer

The ID of the template script.

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

Update some attributes of this CustomScript

Parameters *id* : integer

The ID for the script.

arguments : dict, optional

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

schedule : dict, optional:

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

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

credential_id : integer, optional

The credential that this script will use.

name : string, optional

The name of the script.

target_project_id : integer, optional

Target project to which script outputs will be added.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

Returns `id` : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

last_run : dict:

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

params : list:

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

type : string

The type of the script (e.g Custom)

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

notifications : dict:

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

schedule : dict:

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

finished_at : string/time

The time that the script's last run finished.

code_preview : string

The code that this script will run with arguments inserted.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:

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

author : dict:

```

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

```

template_script_name : string

The name of the template script.

projects : list:

```

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

```

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

credential_id : integer

The credential that this script will use.

name : string

The name of the script.

from_template_id : integer

The ID of the template script.

state : string

The status of the script's last run.

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

Update some attributes of this JavaScript Script

Parameters **id** : integer

The ID for the script.

user_context : string, optional

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

params : list, optional:

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

target_project_id : integer, optional

Target project to which script outputs will be added.

source : string, optional

The body/text of the script.

credential_id : integer, optional

The credential that this script will use.

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.

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

```
- failure_on : boolean
  If failure email notifications are on
- failure_email_addresses : list
  Addresses to notify by e-mail when the job fails.
- success_on : boolean
  If success email notifications are on
- stall_warning_minutes : integer
  Stall warning emails will be sent after this amount of minutes.
- success_email_subject : string
```



```

    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
    ↪ successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.

```

time_zone : string, optional

The time zone of this script.

schedule : dict, optional:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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

```

name : string, optional

The name of the script.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

Returns author : dict:

```

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

```

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

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

params : list:

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

type : string

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

credential_id : integer

The credential that this script will use.

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

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

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:

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

schedule : dict:

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

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

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

source : string

The body/text of the script.

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

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

Update some attributes of this Python Script

Parameters *id* : integer

The ID for the script.

user_context : string, optional

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

params : list, optional:

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

required_resources : dict, optional:

```
- 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.
- cpu : integer
  The number of CPU shares to allocate for the container. Each
↳ core has
  1024 shares. Must be at least 2 shares.
```

source : string, optional

The body/text of the script.

target_project_id : integer, optional

Target project to which script outputs will be added.

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.

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

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

time_zone : string, optional

The time zone of this script.

schedule : dict, optional:

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

name : string, optional

The name of the script.

Returns author : dict:

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

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

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

params : list:

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

type : string

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```
- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
```

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

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

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

running_as : dict:

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

schedule : dict:

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

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

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

source : string

The body/text of the script.

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

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

Update some attributes of this R Script

Parameters **id** : integer

The ID for the script.

user_context : string, optional

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

params : list, optional:

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

required_resources : dict, optional:

```
- 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.
  - cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪ core has
    1024 shares. Must be at least 2 shares.

```

source : string, optional

The body/text of the script.

target_project_id : integer, optional

Target project to which script outputs will be added.

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.

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

```

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

```

time_zone : string, optional

The time zone of this script.

schedule : dict, optional:

```

- scheduled : boolean
  If the object is scheduled
- scheduled_minutes : list
  Minutes of the day it is scheduled on
- scheduled_hours : list

```

```
Hours of the day it is scheduled on
- 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
```

name : string, optional

The name of the script.

Returns **author** : dict:

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

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

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

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- 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.
```

```

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

```

type : string

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```

- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.

```

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

```

- failure_on : boolean
    If failure email notifications are on
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean

```

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

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

running_as : dict:

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

schedule : dict:

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

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

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

source : string

The body/text of the script.

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

patch_sql (*id*, ***kwargs*)

Update some attributes of this SQL script

Parameters id : integer

The ID for the script.

user_context : string, optional

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

params : list, optional:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- 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.
- description : string
    A short sentence or fragment describing this parameter to the,
↳end user.
- required : boolean
    Whether this param is required.
- type : string
    The type of parameter. Valid options: string, integer, float,
↳bool,
    database, credential_aws, credential_redshift, or credential_
↳custom
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
```

target_project_id : integer, optional

Target project to which script outputs will be added.

credential_id : integer, optional

The credential that this script will use.

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.

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

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

time_zone : string, optional

The time zone of this script.

schedule : dict, optional:

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

name : string, optional

The name of the script.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

sql : string, optional

The raw SQL query for the script.

Returns author : dict:

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

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

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

params : list:

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

type : string

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

credential_id : integer

The credential that this script will use.

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

```
- failure_on : boolean  
  If failure email notifications are on  
- failure_email_addresses : list  
  Addresses to notify by e-mail when the job fails.  
- success_on : boolean  
  If success email notifications are on  
- stall_warning_minutes : integer  
  Stall warning emails will be sent after this amount of minutes.  
- success_email_subject : string  
  Custom subject line for success e-mail.
```

```

- urls : list
    URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
    ↪ successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.

```

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.

```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

code_preview : string

The code that this script will run with arguments inserted.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:

```

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

```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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

```

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

projects : list:

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

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

sql : string

The raw SQL query for the script.

post (*credential_id*, *name*, *remote_host_id*, *sql*, ***kwargs*)
Create a script

Parameters credential_id : integer

The credential ID.

name : string

The name of the script.

remote_host_id : integer

The database ID.

sql : string

The raw SQL query for the script.

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.

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.
- 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.
- description : string
    A short sentence or fragment describing this parameter to the_
↳end user.
- required : boolean
    Whether this param is required.
- type : string
    The type of parameter. Valid options: string, integer, float,_
↳bool,
    database, credential_aws, credential_redshift, or credential_
↳custom
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
```

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:

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

hidden : boolean, optional

The hidden status of the object.

Returns **author** : dict:

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

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

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

params : list:

```
A definition of the parameters this script accepts in the arguments.
↪field.
- 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.
- description : string
    A short sentence or fragment describing this parameter to the
↪end user.
- required : boolean
    Whether this param is required.
- type : string
    The type of parameter. Valid options: string, integer, float,
↪bool,
```

```

    database, credential_aws, credential_redshift, or credential_
↪custom
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.

```

type : string

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

template_script_id : integer

The ID of the template script, if any.

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

```

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

```

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.

```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

running_as : dict:

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

schedule : dict:

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

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

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

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

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 (*docker_command*, *docker_image_name*, *required_resources*, ***kwargs*)

Create a container

Parameters **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.

required_resources : dict:

```
- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
↪ core has
    1024 shares.
```

remote_host_credential_id : integer, optional

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

user_context : string, optional

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

repo_ref : string, optional

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

params : list, optional:

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

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.

hidden : boolean, optional

The hidden status of the object.

arguments : dict, optional

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

docker_image_tag : string, optional

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

```

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

```

schedule : dict, optional:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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

```

name : string, optional

The name of the container.

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.

Returns **author** : dict:

```

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

```

id : integer

The ID for the script.

remote_host_credential_id : integer

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

user_context : string

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

docker_command : string

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

params : list:

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

type : string

The type of the script (e.g Container)

last_run : dict:

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```
- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
↪ core has
    1024 shares.
```

archived : string

The archival status of the requested object(s).

notifications : dict:

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

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

repo_http_uri : string

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

running_as : dict:

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

schedule : dict:

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

is_template : boolean

Whether others scripts use this one as a template.

repo_ref : string

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

template_script_name : string

The name of the template script.

docker_image_name : string

The name of the docker image to pull from DockerHub.

projects : list:

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

template_dependents_count : integer

How many other scripts use this one as a template.

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

git_credential_id : integer

The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_image_tag : string

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

name : string

The name of the container.

from_template_id : integer

The ID of the template script.

post_containers_runs (*id*)

Start a run

Parameters id : integer

The ID of the container.

Returns 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.

container_id : integer

The ID of the container.

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.

started_at : string/time

The time the last run started at.

post_custom (*from_template_id, **kwargs*)

Create a CustomScript

Parameters from_template_id : integer

The ID of the template script.

hidden : boolean, optional

The hidden status of the object.

arguments : dict, optional

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

schedule : dict, optional:

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

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

credential_id : integer, optional

The credential that this script will use.

name : string, optional

The name of the script.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

target_project_id : integer, optional

Target project to which script outputs will be added.

Returns id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

last_run : dict:

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

params : list:

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

type : string

The type of the script (e.g Custom)

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

notifications : dict:

```
- failure_on : boolean
    If failure email notifications are on
```

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

schedule : dict:

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

finished_at : string/time

The time that the script's last run finished.

code_preview : string

The code that this script will run with arguments inserted.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:

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

author : dict:

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

```

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

```

template_script_name : string

The name of the template script.

projects : list:

```

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

```

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

credential_id : integer

The credential that this script will use.

name : string

The name of the script.

from_template_id : integer

The ID of the template script.

state : string

The status of the script's last run.

post_custom_runs (*id*)

Start a run

Parameters id : integer

The ID of the custom.

Returns 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.

started_at : string/time

The time the last run started at.

custom_id : integer

The ID of the custom.

post_javascript (*source, credential_id, name, remote_host_id, **kwargs*)

Create a JavaScript Script

Parameters **source** : string

The body/text of the script.

credential_id : integer

The credential that this script will use.

name : string

The name of the script.

remote_host_id : integer

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.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- 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.
- description : string
  A short sentence or fragment describing this parameter to the_
↪end user.
- required : boolean
  Whether this param is required.
- type : string
  The type of parameter. Valid options: string, integer, float,_
↪bool,
  database, credential_aws, credential_redshift, or credential_
↪custom
- name : string
```

```

    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.

```

target_project_id : integer, optional

Target project to which script outputs will be added.

hidden : boolean, optional

The hidden status of the object.

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.

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

```

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

```

time_zone : string, optional

The time zone of this script.

schedule : dict, optional:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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

```

Returns `author` : dict:

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

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

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

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- 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.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- required : boolean
    Whether this param is required.
- type : string
    The type of parameter. Valid options: string, integer, float,_
↪bool,
    database, credential_aws, credential_redshift, or credential_
↪custom
- name : string
```

The variable's name as used within your code.

- label : string
- The label to present to users when asking them **for** the value.

type : string

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

credential_id : integer

The credential that this script will use.

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

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

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.

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:

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

schedule : dict:

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

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

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

source : string

The body/text of the script.

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

post_javascript_runs (*id*)

Start a run

Parameters **id** : integer

The ID of the javascript.

Returns **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.

javascript_id : integer

The ID of the javascript.

started_at : string/time

The time the last run started at.

post_python3 (*source*, *name*, ***kwargs*)

Create a Python Script

Parameters **source** : string

The body/text of the script.

name : string

The name of the script.

user_context : string, optional

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

params : list, optional:

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

required_resources : dict, optional:

```
- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each_
↪core has
    1024 shares. Must be at least 2 shares.
```

target_project_id : integer, optional

Target project to which script outputs will be added.

hidden : boolean, optional

The hidden status of the object.

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.

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

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

time_zone : string, optional

The time zone of this script.

schedule : dict, optional:

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

Returns author : dict:

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

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

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

params : list:

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

type : string

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```
- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has 1024 shares. Must be at least 2 shares.
```

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

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

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

running_as : dict:

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

schedule : dict:

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

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

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

source : string

The body/text of the script.

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

post_python3_runs (*id*)

Start a run

Parameters **id** : integer

The ID of the python.

Returns **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.

started_at : string/time

The time the last run started at.

post_r (*source, name, **kwargs*)

Create an R Script

Parameters **source** : string

The body/text of the script.

name : string

The name of the script.

user_context : string, optional

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

params : list, optional:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- 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.
- description : string
  A short sentence or fragment describing this parameter to the
↳end user.
- required : boolean
  Whether this param is required.
- type : string
  The type of parameter. Valid options: string, integer, float,
↳bool,
database, credential_aws, credential_redshift, or credential_
↳custom
- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.
```

required_resources : dict, optional:

```
- 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.
- cpu : integer
  The number of CPU shares to allocate for the container. Each
↳core has
1024 shares. Must be at least 2 shares.
```

target_project_id : integer, optional

Target project to which script outputs will be added.

hidden : boolean, optional

The hidden status of the object.

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.

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:


```

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

```

time_zone : string, optional

The time zone of this script.

schedule : dict, optional:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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

```

Returns author : dict:

```

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

```

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

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

params : list:

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

type : string

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```
- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
↪core has
    1024 shares. Must be at least 2 shares.

```

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

```

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

```

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.

```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

running_as : dict:

```

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

```

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

schedule : dict:

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

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

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

source : string

The body/text of the script.

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

post_r_runs (*id*)

Start a run

Parameters **id** : integer

The ID of the r.

Returns **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.

r_id : integer

The ID of the r.

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.

started_at : string/time

The time the last run started at.

post_run (*id*)

Run a script

Parameters **id** : integer

The ID for the script.

Returns None

Response code 204: success

post_sql (*credential_id, name, remote_host_id, sql, **kwargs*)

Create a SQL script

Parameters **credential_id** : integer

The credential that this script will use.

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.

user_context : string, optional

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

params : list, optional:

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

target_project_id : integer, optional

Target project to which script outputs will be added.

hidden : boolean, optional

The hidden status of the object.

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.

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

```
- failure_on : boolean
  If failure email notifications are on
- failure_email_addresses : list
  Addresses to notify by e-mail when the job fails.
- success_on : boolean
  If success email notifications are on
- stall_warning_minutes : integer
```

```

    Stall warning emails will be sent after this amount of minutes.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
↳ successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.

```

time_zone : string, optional

The time zone of this script.

schedule : dict, optional:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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

```

Returns author : dict:

```

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

```

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

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

last_run : dict:

```

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

params : list:

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

type : string

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

credential_id : integer

The credential that this script will use.

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

```
- failure_on : boolean
    If failure email notifications are on
- failure_email_addresses : list
```



```

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

```

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.

```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

code_preview : string

The code that this script will run with arguments inserted.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:

```

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

```

schedule : dict:

```

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

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

projects : list:

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

from_template_id : integer

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

sql : string

The raw SQL query for the script.

post_sql_runs (*id*)

Start a run

Parameters id : integer

The ID of the sql.

Returns `sql_id` : integer

The ID of this sql.

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.

state : string

The state of this run.

finished_at : string/time

The time that this run finished.

output : list:

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

started_at : string/time

The time the last run started.

put_containers (*id*, *docker_command*, *docker_image_name*, *required_resources*, ***kwargs*)

Edit a container

Parameters `id` : integer

The ID for the script.

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.

required_resources : dict:

```
- 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.
- cpu : integer
```

```
The number of CPU shares to allocate for the container. Each_
↳core has
  1024 shares.
```

remote_host_credential_id : integer, optional

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

user_context : string, optional

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

repo_ref : string, optional

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

params : list, optional:

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

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.

arguments : dict, optional

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

docker_image_tag : string, optional

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

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

schedule : dict, optional:

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

name : string, optional

The name of the container.

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.

Returns author : dict:

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

id : integer

The ID for the script.

remote_host_credential_id : integer

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

user_context : string

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

docker_command : string

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

params : list:

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

type : string

The type of the script (e.g Container)

last_run : dict:

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```
- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
↪core has
    1024 shares.
```

archived : string

The archival status of the requested object(s).

notifications : dict:

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

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

repo_http_uri : string

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

running_as : dict:

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

schedule : dict:

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

is_template : boolean

Whether others scripts use this one as a template.

repo_ref : string

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

template_script_name : string

The name of the template script.

docker_image_name : string

The name of the docker image to pull from DockerHub.

projects : list:

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

template_dependents_count : integer

How many other scripts use this one as a template.

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

git_credential_id : integer

The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_image_tag : string

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

name : string

The name of the container.

from_template_id : integer

The ID of the template script.

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 author : dict:

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

id : integer

The ID for the script.

remote_host_credential_id : integer

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

user_context : string

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

docker_command : string

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

params : list:

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

type : string

The type of the script (e.g Container)

last_run : dict:

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

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```

- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
↪core has
    1024 shares.

```

archived : string

The archival status of the requested object(s).

notifications : dict:

```

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

```

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.

```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

repo_http_uri : string

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

running_as : dict:

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

schedule : dict:

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

is_template : boolean

Whether others scripts use this one as a template.

repo_ref : string

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

template_script_name : string

The name of the template script.

docker_image_name : string

The name of the docker image to pull from DockerHub.

projects : list:

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

template_dependents_count : integer

How many other scripts use this one as a template.

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

git_credential_id : integer

The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.

docker_image_tag : string

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

name : string

The name of the container.

from_template_id : integer

The ID of the template script.

put_containers_projects (*id, project_id*)

Add a container docker to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_containers_shares_groups (*id, group_ids, permission_level*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

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

Returns **readers** : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

put_containers_shares_users (*id*, *permission_level*, *user_ids*)

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

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

user_ids : list

An array of one or more user IDs

Returns **readers** : dict:

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

total_group_shares : integer

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

owners : dict:

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

writers : dict:

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

put_custom(*id*, ****kwargs**)

Replace all attributes of this CustomScript

Parameters *id* : integer

The ID for the script.

arguments : dict, optional

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

schedule : dict, optional:

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

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

credential_id : integer, optional

The credential that this script will use.

name : string, optional

The name of the script.

target_project_id : integer, optional

Target project to which script outputs will be added.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

Returns id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

last_run : dict:

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

params : list:

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


type : string

The type of the script (e.g Custom)

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

notifications : dict:

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

schedule : dict:

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

finished_at : string/time

The time that the script's last run finished.

code_preview : string

The code that this script will run with arguments inserted.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:

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

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

author : dict:

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

template_script_name : string

The name of the template script.

projects : list:

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

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

credential_id : integer

The credential that this script will use.

name : string

The name of the script.

from_template_id : integer

The ID of the template script.

state : string

The status of the script's last run.

put_custom_archive (*id*, *status*)

Update the archive status of this object

Parameters *id* : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns *id* : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

last_run : dict:

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

params : list:

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

type : string

The type of the script (e.g Custom)

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

notifications : dict:

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

schedule : dict:

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

finished_at : string/time

The time that the script's last run finished.

code_preview : string

The code that this script will run with arguments inserted.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:

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

```

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

```

author : dict:

```

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

```

template_script_name : string

The name of the template script.

projects : list:

```

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

```

hidden : boolean

The hidden status of the object.

arguments : dict

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

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

credential_id : integer

The credential that this script will use.

name : string

The name of the script.

from_template_id : integer

The ID of the template script.

state : string

The status of the script's last run.

put_custom_projects (*id*, *project_id*)

Add a Job to a project

Parameters *id* : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_custom_shares_groups (*id*, *group_ids*, *permission_level*)

Set the permissions groups has on this object

Parameters *id* : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

Returns *readers* : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

owners : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : 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.

writers : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
```

```
- id : integer
- name : string
```

put_custom_shares_users (*id*, *permission_level*, *user_ids*)

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

permission_level : string

Options are: “read”, “write”, or “manage”

user_ids : list

An array of one or more user IDs

Returns *readers* : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

owners : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : 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.

writers : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

put_javascript (*id*, *source*, *credential_id*, *name*, *remote_host_id*, ***kwargs*)

Replace all attributes of this JavaScript Script

Parameters *id* : integer

The ID for the script.

source : string

The body/text of the script.

credential_id : integer

The credential that this script will use.

name : string

The name of the script.

remote_host_id : integer

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.

params : list, optional:

A definition of the parameters this script accepts **in** the arguments.

- ↪ **field**.
- **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**.
- **description** : string
 - A short sentence **or** fragment describing this parameter to the
 - ↪ **end user**.
- **required** : boolean
 - Whether this param **is** required.
- **type** : string
 - The **type** of parameter. Valid options: string, integer, float,
 - ↪ **bool**,
 - database, credential_aws, credential_redshift, **or** credential_
 - ↪ **custom**
- **name** : string
 - The variable's **name** as used within your code.
- **label** : string
 - The label to present to users when asking them **for** the value.

target_project_id : integer, optional

Target project to which script outputs will be added.

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.

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:


```

- failure_on : boolean
    If failure email notifications are on
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes.
↪successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.

```

time_zone : string, optional

The time zone of this script.

schedule : dict, optional:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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

```

Returns author : dict:

```

- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
    This user's name.
- online : boolean
    Whether this user is online.

```

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

last_run : dict:

```
- 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.
- started_at : string/time
    The time that the run started.
```

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- 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.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- required : boolean
    Whether this param is required.
- type : string
    The type of parameter. Valid options: string, integer, float,
↳bool,
    database, credential_aws, credential_redshift, or credential_
↳custom
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
```

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

credential_id : integer

The credential that this script will use.

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

```
- failure_on : boolean
    If failure email notifications are on
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
↳ successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
```

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:

```
- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
    This user's name.
- online : boolean
    Whether this user is online.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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
```

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

```
A list of projects containing the script.
- id : integer
  The ID for the project.
- name : string
  The name of the project.
```

source : string

The body/text of the script.

from_template_id : integer

The ID of the template this script uses, if any.

hidden : boolean

The hidden status of the object.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

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 `author` : dict:

```
- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
    This user's name.
- online : boolean
    Whether this user is online.
```

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

last_run : dict:

```
- 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.
- started_at : string/time
    The time that the run started.
```

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- 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.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- required : boolean
    Whether this param is required.
- type : string
    The type of parameter. Valid options: string, integer, float,_
↪bool,
    database, credential_aws, credential_redshift, or credential_
↪custom
```

```
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
```

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

credential_id : integer

The credential that this script will use.

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

```
- failure_on : boolean
    If failure email notifications are on
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
```

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:

```
- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
    This user's name.
- online : boolean
    Whether this user is online.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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
```

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

source : string

The body/text of the script.

from_template_id : integer

The ID of the template this script uses, if any.

hidden : boolean

The hidden status of the object.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

put_javascript_projects (*id*, *project_id*)

Add a scripted sql to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_javascript_shares_groups (*id*, *group_ids*, *permission_level*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

Returns **readers** : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

owners : dict:


```
- groups : list::
  - id : integer
  - name : string
- users : 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.

writers : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

put_javascript_shares_users (*id*, *permission_level*, *user_ids*)

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

Options are: “read”, “write”, or “manage”

user_ids : list

An array of one or more user IDs

Returns **readers** : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

owners : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : 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.

writers : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

put_python3 (*id*, *source*, *name*, ***kwargs*)
Replace all attributes of this Python Script

Parameters *id* : integer

The ID for the script.

source : string

The body/text of the script.

name : string

The name of the script.

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- 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.
- description : string
  A short sentence or fragment describing this parameter to the_
↪end user.
- required : boolean
  Whether this param is required.
- type : string
  The type of parameter. Valid options: string, integer, float,_
↪bool,
  database, credential_aws, credential_redshift, or credential_
↪custom
- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.
```

required_resources : dict, optional:

```
- 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.
- cpu : integer
  The number of CPU shares to allocate for the container. Each
↪core has
  1024 shares. Must be at least 2 shares.

```

target_project_id : integer, optional

Target project to which script outputs will be added.

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.

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

```

- failure_on : boolean
  If failure email notifications are on
- failure_email_addresses : list
  Addresses to notify by e-mail when the job fails.
- success_on : boolean
  If success email notifications are on
- stall_warning_minutes : integer
  Stall warning emails will be sent after this amount of minutes.
- success_email_subject : string
  Custom subject line for success e-mail.
- urls : list
  URLs to receive a POST request at job completion
- success_email_addresses : list
  Addresses to notify by e-mail when the job completes
↪successfully.
- success_email_body : string
  Custom body text for success e-mail, written in Markdown.

```

time_zone : string, optional

The time zone of this script.

schedule : dict, optional:

```

- scheduled : boolean
  If the object is scheduled
- scheduled_minutes : list
  Minutes of the day it is scheduled on

```

```
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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
```

Returns **author** : dict:

```
- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
    This user's name.
- online : boolean
    Whether this user is online.
```

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

last_run : dict:

```
- 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.
- started_at : string/time
    The time that the run started.
```

params : list:

```
A definition of the parameters this script accepts in the arguments_
↳field.
- 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.
- description : string
    A short sentence or fragment describing this parameter to the_
↳end user.
```

```

- required : boolean
    Whether this param is required.
- type : string
    The type of parameter. Valid options: string, integer, float, bool,
    database, credential_aws, credential_redshift, or credential_
    custom
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.

```

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```

- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    core has 1024 shares. Must be at least 2 shares.

```

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

```

- failure_on : boolean
    If failure email notifications are on
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.

```

```
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes,
↳ successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
```

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

running_as : dict:

```
- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
    This user's name.
- online : boolean
    Whether this user is online.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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
```

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

source : string

The body/text of the script.

from_template_id : integer

The ID of the template this script uses, if any.

hidden : boolean

The hidden status of the object.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

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 **author** : dict:

```
- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
    This user's name.
- online : boolean
    Whether this user is online.
```

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

last_run : dict:

```
- 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.
- started_at : string/time
    The time that the run started.
```

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- 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.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- required : boolean
    Whether this param is required.
- type : string
    The type of parameter. Valid options: string, integer, float,
↳bool,
    database, credential_aws, credential_redshift, or credential_
↳custom
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
```

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```
- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
```

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

```
- failure_on : boolean
    If failure email notifications are on
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes
    ↪successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
```

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

running_as : dict:

```
- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
    This user's name.
- online : boolean
    Whether this user is online.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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
```

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

source : string

The body/text of the script.

from_template_id : integer

The ID of the template this script uses, if any.

hidden : boolean

The hidden status of the object.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

put_python3_projects (*id, project_id*)

Add a python docker to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_python3_shares_groups (*id, group_ids, permission_level*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

Returns **readers** : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

owners : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : 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.

writers : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

put_python3_shares_users (*id*, *permission_level*, *user_ids*)

Set the permissions users have on this object

Parameters *id* : integer

ID of the resource to be shared

permission_level : string

Options are: “read”, “write”, or “manage”

user_ids : list

An array of one or more user IDs

Returns **readers** : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

owners : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : 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.

writers : dict:

```

- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string

```

put_r (*id*, *source*, *name*, ***kwargs*)

Replace all attributes of this R Script

Parameters *id* : integer

The ID for the script.

source : string

The body/text of the script.

name : string

The name of the script.

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

```

A definition of the parameters this script accepts in the arguments.
↳field.
- 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.
- description : string
  A short sentence or fragment describing this parameter to the
  ↳end user.
- required : boolean
  Whether this param is required.
- type : string
  The type of parameter. Valid options: string, integer, float,
  ↳bool,
  database, credential_aws, credential_redshift, or credential_
  ↳custom
- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.

```

required_resources : dict, optional:

```

- 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.
- cpu : integer
  The number of CPU shares to allocate for the container. Each_
↪core has
1024 shares. Must be at least 2 shares.
```

target_project_id : integer, optional

Target project to which script outputs will be added.

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.

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

```
- failure_on : boolean
  If failure email notifications are on
- failure_email_addresses : list
  Addresses to notify by e-mail when the job fails.
- success_on : boolean
  If success email notifications are on
- stall_warning_minutes : integer
  Stall warning emails will be sent after this amount of minutes.
- success_email_subject : string
  Custom subject line for success e-mail.
- urls : list
  URLs to receive a POST request at job completion
- success_email_addresses : list
  Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_body : string
  Custom body text for success e-mail, written in Markdown.
```

time_zone : string, optional

The time zone of this script.

schedule : dict, optional:

```
- scheduled : boolean
  If the object is scheduled
- scheduled_minutes : list
  Minutes of the day it is scheduled on
- scheduled_hours : list
  Hours of the day it is scheduled on
- 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
```

Returns `author` : dict:

```
- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
    This user's name.
- online : boolean
    Whether this user is online.
```

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

last_run : dict:

```
- 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.
- started_at : string/time
    The time that the run started.
```

params : list:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- 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.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- required : boolean
    Whether this param is required.
- type : string
```

```
The type of parameter. Valid options: string, integer, float, bool,  
database, credential_aws, credential_redshift, or credential_  
custom  
- name : string  
  The variable's name as used within your code.  
- label : string  
  The label to present to users when asking them for the value.
```

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```
- 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.  
- cpu : integer  
  The number of CPU shares to allocate for the container. Each  
core has 1024 shares. Must be at least 2 shares.
```

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

```
- failure_on : boolean  
  If failure email notifications are on  
- failure_email_addresses : list  
  Addresses to notify by e-mail when the job fails.  
- success_on : boolean  
  If success email notifications are on  
- stall_warning_minutes : integer  
  Stall warning emails will be sent after this amount of minutes.  
- success_email_subject : string  
  Custom subject line for success e-mail.  
- urls : list
```



```

    URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.

```

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.

```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

running_as : dict:

```

- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
    This user's name.
- online : boolean
    Whether this user is online.

```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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

```

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

```

A list of projects containing the script.
- id : integer

```

```
The ID for the project.  
- name : string  
  The name of the project.
```

source : string

The body/text of the script.

from_template_id : integer

The ID of the template this script uses, if any.

hidden : boolean

The hidden status of the object.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

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 **author** : dict:

```
- initials : string  
  This user's initials.  
- username : string  
  This user's username.  
- id : integer  
  The ID of this user.  
- name : string  
  This user's name.  
- online : boolean  
  Whether this user is online.
```

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

last_run : dict:

```
- 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.
- started_at : string/time
    The time that the run started.
```

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- 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.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- required : boolean
    Whether this param is required.
- type : string
    The type of parameter. Valid options: string, integer, float,
↳bool,
    database, credential_aws, credential_redshift, or credential_
↳custom
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
```

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

required_resources : dict:

```
- 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.
- cpu : integer
    The number of CPU shares to allocate for the container. Each
    ↪core has
    1024 shares. Must be at least 2 shares.
```

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

```
- failure_on : boolean
    If failure email notifications are on
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes
    ↪successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
```

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

running_as : dict:

```

- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
    This user's name.
- online : boolean
    Whether this user is online.

```

schedule : dict:

```

- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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

```

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

projects : list:

```

A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.

```

source : string

The body/text of the script.

from_template_id : integer

The ID of the template this script uses, if any.

hidden : boolean

The hidden status of the object.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

put_r_projects (*id*, *project_id*)

Add a r docker to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_r_shares_groups (*id*, *group_ids*, *permission_level*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

Returns **readers** : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

owners : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : 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.

writers : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

put_r_shares_users (*id*, *permission_level*, *user_ids*)

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

permission_level : string

Options are: “read”, “write”, or “manage”

user_ids : list

An array of one or more user IDs

Returns **readers** : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

owners : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : 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.

writers : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
```

```
- id : integer
- name : string
```

put_sql (*id*, *credential_id*, *name*, *remote_host_id*, *sql*, ***kwargs*)

Replace all attributes of this SQL script

Parameters *id* : integer

The ID for the script.

credential_id : integer

The credential that this script will use.

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.

user_context : string, optional

“runner” or “author”, who to execute the script as when run as a template.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments_
↪field.
- 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.
- description : string
  A short sentence or fragment describing this parameter to the_
↪end user.
- required : boolean
  Whether this param is required.
- type : string
  The type of parameter. Valid options: string, integer, float,_
↪bool,
  database, credential_aws, credential_redshift, or credential_
↪custom
- name : string
  The variable's name as used within your code.
- label : string
  The label to present to users when asking them for the value.
```

target_project_id : integer, optional

Target project to which script outputs will be added.

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.

parent_id : integer, optional

The ID of the parent job that will trigger this script

notifications : dict, optional:

```
- failure_on : boolean
    If failure email notifications are on
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↳successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
```

time_zone : string, optional

The time zone of this script.

schedule : dict, optional:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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
```

Returns author : dict:

```
- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
    This user's name.
- online : boolean
    Whether this user is online.
```

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

last_run : dict:

```
- 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.
- started_at : string/time
    The time that the run started.
```

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳field.
- 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.
- description : string
    A short sentence or fragment describing this parameter to the
↳end user.
- required : boolean
    Whether this param is required.
- type : string
    The type of parameter. Valid options: string, integer, float,
↳bool,
    database, credential_aws, credential_redshift, or credential_
↳custom
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
```

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

credential_id : integer

The credential that this script will use.

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

```
- failure_on : boolean
    If failure email notifications are on
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes
    successfully.
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
```

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

code_preview : string

The code that this script will run with arguments inserted.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:

```
- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
```

```
This user's name.  
- online : boolean  
  Whether this user is online.
```

schedule : dict:

```
- scheduled : boolean  
  If the object is scheduled  
- scheduled_minutes : list  
  Minutes of the day it is scheduled on  
- scheduled_hours : list  
  Hours of the day it is scheduled on  
- 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
```

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

projects : list:

```
A list of projects containing the script.  
- id : integer  
  The ID for the project.  
- name : string  
  The name of the project.
```

from_template_id : integer

The ID of the template this script uses, if any.

hidden : boolean

The hidden status of the object.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

sql : string

The raw SQL query for the script.

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 **author** : dict:

```
- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
    This user's name.
- online : boolean
    Whether this user is online.
```

id : integer

The ID for the script.

archived : string

The archival status of the requested object(s).

user_context : string

“runner” or “author”, who to execute the script as when run as a template.

last_run : dict:

```
- 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.
- started_at : string/time
    The time that the run started.
```

params : list:

```
A definition of the parameters this script accepts in the arguments.
↳ field.
- 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.
- description : string
    A short sentence or fragment describing this parameter to the_
↪end user.
- required : boolean
    Whether this param is required.
- type : string
    The type of parameter. Valid options: string, integer, float, _
↪bool,
    database, credential_aws, credential_redshift, or credential_
↪custom
- name : string
    The variable's name as used within your code.
- label : string
    The label to present to users when asking them for the value.
```

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

updated_at : string/time

The time the script was last updated.

target_project_id : integer

Target project to which script outputs will be added.

published_as_template_id : integer

The ID of the template that this script is backing.

credential_id : integer

The credential that this script will use.

next_run_at : string/time

The time of the next scheduled run.

notifications : dict:

```
- failure_on : boolean
    If failure email notifications are on
- failure_email_addresses : list
    Addresses to notify by e-mail when the job fails.
- success_on : boolean
    If success email notifications are on
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes_
↪successfully.
```

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
```

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.
```

state : string

The status of the script's last run.

finished_at : string/time

The time that the script's last run finished.

code_preview : string

The code that this script will run with arguments inserted.

remote_host_id : integer

The remote host ID that this script will connect to.

running_as : dict:

```
- initials : string
    This user's initials.
- username : string
    This user's username.
- id : integer
    The ID of this user.
- name : string
    This user's name.
- online : boolean
    Whether this user is online.
```

schedule : dict:

```
- scheduled : boolean
    If the object is scheduled
- scheduled_minutes : list
    Minutes of the day it is scheduled on
- scheduled_hours : list
    Hours of the day it is scheduled on
- 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
```

is_template : boolean

Whether others scripts use this one as a template.

template_script_name : string

The name of the template script.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

projects : list:

A **list** of projects containing the script.

- **id** : integer
The ID **for** the project.
- **name** : string
The name of the project.

from_template_id : integer

The ID of the template this script uses, if any.

hidden : boolean

The hidden status of the object.

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

created_at : string/time

The time this script was created.

parent_id : integer

The ID of the parent job that will trigger this script

time_zone : string

The time zone of this script.

name : string

The name of the script.

template_dependents_count : integer

How many other scripts use this one as a template.

sql : string

The raw SQL query for the script.

put_sql_projects (*id*, *project_id*)

Add a scripts to a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

put_sql_shares_groups (*id*, *group_ids*, *permission_level*)

Set the permissions groups has on this object

Parameters **id** : integer

ID of the resource to be shared

group_ids : list

An array of one or more group IDs

permission_level : string

Options are: “read”, “write”, or “manage”

Returns readers : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

owners : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : 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.

writers : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

put_sql_shares_users (*id*, *permission_level*, *user_ids*)

Set the permissions users have on this object

Parameters id : integer

ID of the resource to be shared

permission_level : string

Options are: “read”, “write”, or “manage”

user_ids : list

An array of one or more user IDs

Returns readers : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

total_group_shares : integer

For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

owners : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : 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.

writers : dict:

```
- groups : list::
  - id : integer
  - name : string
- users : list::
  - id : integer
  - name : string
```

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(schema, data, name, database_id)</code>	Import a file into a table

Continued on next page

Table 4.14 – continued from previous page

<code>post_enhancements_cass_ncoa(source_table_id, Standardize addresses in a table ...)</code>	
<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** *id* : integer

The ID of the table.

is_view : boolean

True if this table represents a view. False if it represents a regular table.

owner : string

The database username of the table's owner.

last_run : dict:

```
- 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.
- started_at : string/time
    The time that the run started.
```

enhancements : list:

```
- join_id : integer
- type : string
- updated_at : string/time
- created_at : string/time
```

distkey : string

The column used as the Amazon Redshift distkey.

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.

refresh_id : string

The ID of the most recent statistics refresh.

size_mb : number/float

The size of the table in megabytes.

columns : list:

```
- sql_type : string
    SQL type of 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.
- sample_values : list
    A sample of values from the column.
- useable_as_primary_key : boolean
    Whether the column may be used as an primary key to identify
    ↪ table
    rows.
- order : integer
    Relative position of the column in the table.
- max_value : string
    Largest value in the column.
- encoding : string
    The compression encoding for this columnSee: http://docs.aws.amazon.com/redshift/latest/dg/c\_Compression\_encodings.html
- useable_as_independent_variable : boolean
    Whether the column may be used as an independent variable to
    ↪ train a
    model.
- stddev : number/float
    Stddev of the column, where applicable.
- coverage_count : integer
    Number of non-null values in the column.
- min_value : string
    Smallest value in the column.
- distinct_count : integer
    Number of distinct values in 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.
- avg_value : number/float
    Average value of the column, where applicable.
- value_distribution : dict
    An object mapping distinct values in the column to the number
    ↪ of times
    they appear in the column
- description : string
    The description of the column, as specified by the table owner
- name : string
    Name of the column.
- null_count : integer
    Number of null values in the column.
```

outgoing_table_matches : list:

```
- source_table_id : integer
    Source table
- job : dict::
```

```

- id : integer
- created_at : string/date-time
- last_run : dict::
  - 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.
  - started_at : string/time
    The time that the run started.
- state : string
  Whether the job is idle, queued, running, cancelled, or
↳ failed.
- hidden : boolean
  The hidden status of the object.
- type : string
- updated_at : string/date-time
- name : string
- match_options : dict::
  - max_matches : integer
  - threshold : string
- runs : list::
  Information about the most recent runs of the job.
  - 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.
  - started_at : string/time
    The time that the run started.
- target_type : string
  Target type
- target_id : integer
  Target ID
- target : dict::
  - name : string

```

sortkeys : string

The column used as the Amazon Redshift sortkey.

last_refresh : string/date-time

The time of the last statistics refresh.

database_id : integer

The ID of the database.

multipart_key : list

joins : list:

```
- left_identifier : string
- id : integer
- left_table_id : integer
- right_identifier : string
- left_join : boolean
- on : string
- updated_at : string/time
- right_table_id : integer
- created_at : string/time
```

ontology_mapping : dict

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

view_def : string

row_count : integer

The number of rows in the table.

description : string

The description of the table, as specified by the table owner

schema : string

The name of the schema containing the table.

column_count : integer

The number of columns in the table.

name : string

Name of the table.

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.

id : integer

The ID of the enhancement.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

ncoa_credential_id : integer

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_name : string

The name of the table created by the enhancement.

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.

perform_ncoa : boolean

Whether to update addresses for records matching the National Change of Address (NCOA) database.

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 **source_table_id** : integer

The ID of the table that was enhanced.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

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.

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.

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.

id : integer

The ID of the enhancement.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match.
Must be less than or equal to 1 and greater than or equal to 0.

match_table_id : integer

The ID of the Dynamo table to match against.

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'.

enhanced_table_name : string

The name of the table created by the enhancement.

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.

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.

id : integer

The ID of the enhancement.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match.
Must be less than or equal to 1 and greater than or equal to 0.

match_table_id : integer

The ID of the Redshift table to match against.

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'.

enhanced_table_name : string

The name of the table created by the enhancement.

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 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 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 **last_refresh** : string/date-time

The time of the last statistics refresh.

id : integer

The ID of the table.

database_id : integer

The ID of the database.

owner : string

The database username of the table’s owner.

last_run : dict:

```
- 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.
- started_at : string/time
    The time that the run started.
```

row_count : integer

The number of rows in the table.

distkey : string

The column used as the Amazon Redshift distkey.

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.

refresh_id : string

The ID of the most recent statistics refresh.

size_mb : number/float

The size of the table in megabytes.

is_view : boolean

True if this table represents a view. False if it represents a regular table.

description : string

The description of the table, as specified by the table owner

schema : string

The name of the schema containing the table.

column_count : integer

The number of columns in the table.

name : string

Name of the table.

sortkeys : string

The column used as the Amazon Redshift sortkey.

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.

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.

sample_values : list

A sample of values from the column.

useable_as_primary_key : boolean

Whether the column may be used as an primary key to identify table rows.

order : integer

Relative position of the column in the table.

max_value : string

Largest value in the column.

encoding : string

The compression encoding for this columnSee: http://docs.aws.amazon.com/redshift/latest/dg/c_Compression_encodings.html

useable_as_independent_variable : boolean

Whether the column may be used as an independent variable to train a model.

stddev : number/float

Stddev of the column, where applicable.

coverage_count : integer

Number of non-null values in the column.

min_value : string

Smallest value in the column.

distinct_count : integer

Number of distinct values in 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.

avg_value : number/float

Average value of the column, where applicable.

value_distribution : dict

An object mapping distinct values in the column to the number of times they appear in the column

description : string

The description of the column, as specified by the table owner

name : string

Name of the column.

null_count : integer

Number of null values in the column.

patch (*id*, ***kwargs*)

Update a table

Parameters **id** : integer

The ID of the table.

ontology_mapping : dict, optional

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

description : string, optional

The user-defined description of the table.

Returns **last_refresh** : string/date-time

The time of the last statistics refresh.

id : integer

The ID of the table.

database_id : integer

The ID of the database.

owner : string

The database username of the table's owner.

last_run : dict:

```
- 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.
```

```
- started_at : string/time
    The time that the run started.
```

row_count : integer

The number of rows in the table.

distkey : string

The column used as the Amazon Redshift distkey.

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.

refresh_id : string

The ID of the most recent statistics refresh.

size_mb : number/float

The size of the table in megabytes.

is_view : boolean

True if this table represents a view. False if it represents a regular table.

description : string

The description of the table, as specified by the table owner

schema : string

The name of the schema containing the table.

column_count : integer

The number of columns in the table.

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.

sortkeys : string

The column used as the Amazon Redshift sortkey.

post (*schema*, *data*, *name*, *database_id*)

Import a file into a table

Parameters **schema** : string

The destination schema name.

data : string

The file to import, uploaded using HTTP multipart.

name : string

The destination table name, without the schema prefix.

database_id : integer

The ID of the destination database.

Returns `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.

started_at : string/date-time

The start time of the last run.

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.

ncoa_credential_id : integer, optional

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

output_level : string, optional

The set of fields persisted by a CASS or NCOA enhancement. For CASS enhancements, one of 'cass' or 'all.' For NCOA enhancements, one of 'cass', 'ncoa', 'coalesced' or 'all'. By default, all fields will be returned.

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.

id : integer

The ID of the enhancement.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

ncoa_credential_id : integer

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_name : string

The name of the table created by the enhancement.

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.

perform_ncoa : boolean

Whether to update addresses for records matching the National Change of Address (NCOA) database.

post_enhancements_geocodings (*source_table_id*)

Geocode a table

Parameters **source_table_id** : integer

The ID of the table to be enhanced.

Returns **source_table_id** : integer

The ID of the table that was enhanced.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

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.

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.

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.

id : integer

The ID of the enhancement.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match. Must be less than or equal to 1 and greater than or equal to 0.

match_table_id : integer

The ID of the Dynamo table to match against.

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'.

enhanced_table_name : string

The name of the table created by the enhancement.

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.

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.

id : integer

The ID of the enhancement.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match. Must be less than or equal to 1 and greater than or equal to 0.

match_table_id : integer

The ID of the Redshift table to match against.

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'.

enhanced_table_name : string

The name of the table created by the enhancement.

post_refresh (*id*)

Request a refresh for column and table statistics

Parameters **id** : integer

Returns **id** : integer

The ID of the table.

is_view : boolean

True if this table represents a view. False if it represents a regular table.

owner : string

The database username of the table's owner.

last_run : dict:

```
- 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.
- started_at : string/time
  The time that the run started.
```

enhancements : list:

```
- join_id : integer
- type : string
- updated_at : string/time
- created_at : string/time
```

distkey : string

The column used as the Amazon Redshift distkey.

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.

refresh_id : string

The ID of the most recent statistics refresh.

size_mb : number/float

The size of the table in megabytes.

columns : list:

```
- sql_type : string
    SQL type of 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.
- sample_values : list
    A sample of values from the column.
- useable_as_primary_key : boolean
    Whether the column may be used as an primary key to identify
    ↪ table
    rows.
- order : integer
    Relative position of the column in the table.
- max_value : string
    Largest value in the column.
- encoding : string
    The compression encoding for this columnSee: http://docs.aws.
    ↪ amazon.com
    /redshift/latest/dg/c_Compression_encodings.html
- useable_as_independent_variable : boolean
    Whether the column may be used as an independent variable to
    ↪ train a
    model.
- stddev : number/float
    Stddev of the column, where applicable.
- coverage_count : integer
    Number of non-null values in the column.
- min_value : string
    Smallest value in the column.
- distinct_count : integer
    Number of distinct values in 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.
- avg_value : number/float
    Average value of the column, where applicable.
- value_distribution : dict
    An object mapping distinct values in the column to the number
    ↪ of times
    they appear in the column
- description : string
    The description of the column, as specified by the table owner
- name : string
    Name of the column.
- null_count : integer
    Number of null values in the column.
```

outgoing_table_matches : list:

```

- source_table_id : integer
    Source table
- job : dict::
    - id : integer
    - created_at : string/date-time
    - last_run : dict::
        - 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.
        - started_at : string/time
            The time that the run started.
    - state : string
        Whether the job is idle, queued, running, cancelled, or
    ↪ failed.
    - hidden : boolean
        The hidden status of the object.
    - type : string
    - updated_at : string/date-time
    - name : string
    - match_options : dict::
        - max_matches : integer
        - threshold : string
    - runs : list::
        Information about the most recent runs of the job.
        - 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.
        - started_at : string/time
            The time that the run started.
- target_type : string
    Target type
- target_id : integer
    Target ID
- target : dict::
    - name : string

```

sortkeys : string

The column used as the Amazon Redshift sortkey.

last_refresh : string/date-time

The time of the last statistics refresh.

database_id : integer

The ID of the database.

multipart_key : list

joins : list:

```
- left_identifier : string
- id : integer
- left_table_id : integer
- right_identifier : string
- left_join : boolean
- on : string
- updated_at : string/time
- right_table_id : integer
- created_at : string/time
```

ontology_mapping : dict

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

view_def : string

row_count : integer

The number of rows in the table.

description : string

The description of the table, as specified by the table owner

schema : string

The name of the schema containing the table.

column_count : integer

The number of columns in the table.

name : string

Name of the table.

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>list_me_api_accesses(**kwargs)</code>	Show API accesses made by or on behalf of the logged-in user
<code>patch_me(**kwargs)</code>	Update info about the logged-in user
<code>post_api_keys(id, name, expires_in, **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 `last_used_at` : string/date-time

The date and time when the key was last used.

scopes : list

The scopes which the key is permissioned on.

id : integer

The ID of the API key.

expires_at : string/date-time

The date and time when the key expired.

created_at : string/date-time

The date and time when the key was created.

expired : boolean

True if the key has expired.

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.

use_count : integer

The number of times the key has been used.

constraints : list:

```
Constraints on the abilities of the created key
- get_allowed : boolean
    Whether the constraint allows GET requests.
- head_allowed : boolean
    Whether the constraint allows HEAD requests.
- delete_allowed : boolean
    Whether the constraint allows DELETE requests.
- patch_allowed : boolean
    Whether the constraint allows PATCH requests.
- post_allowed : boolean
    Whether the constraint allows POST 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 (*id*)

Show info about a user

Parameters **id** : integer

The ID of this user.

Returns **city** : string

The city of this user.

id : integer

The ID of this user.

title : string

The title of this user.

active : string

The account status of this user.

primary_group_id : integer

The ID of the primary group of this user.

github_username : string

The GitHub username of this user.

groups : list:

An array of **all** the groups this user **is in**.

- **id** : integer
The ID of this group.
- **name** : string
The name of this group.
- **organization_id** : integer
The organization associated **with** this group.

email : string

The email of this user.

time_zone : string

The time zone of this user.

prefers_sms_otp : string

The preference for phone authorization of this user

phone : string

The phone number of this user.

user : string

The username of this user.

initials : string

The initials of this user.

state : string

The state of this user.

name : string

The name of this user.

department : string

The department 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 **last_used_at** : string/date-time

The date and time when the key was last used.

scopes : list

The scopes which the key is permissioned on.

id : integer

The ID of the API key.

expires_at : string/date-time

The date and time when the key expired.

created_at : string/date-time

The date and time when the key was created.

expired : boolean

True if the key has expired.

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.

use_count : integer

The number of times the key has been used.

constraints : list:

```
Constraints on the abilities of the created key
- get_allowed : boolean
    Whether the constraint allows GET requests.
- head_allowed : boolean
    Whether the constraint allows HEAD requests.
- delete_allowed : boolean
    Whether the constraint allows DELETE requests.
- patch_allowed : boolean
```

```
Whether the constraint allows PATCH requests.
- post_allowed : boolean
  Whether the constraint allows POST 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).
```

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.

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 **email** : string

The email of this user.

id : integer

The ID of this user.

created_at : string/date-time

The date and time when the user was created.

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.

user : string

The username of this user.

name : string

The name of this user.

groups : list:

An array of **all** the groups this user **is in**.

- **id** : integer
The ID of this group.
- **name** : string
The name of this group.
- **organization_id** : integer
The organization associated **with** this group.

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 **last_used_at** : string/date-time

The date and time when the key was last used.

scopes : list

The scopes which the key is permissioned on.

id : integer

The ID of the API key.

expires_at : string/date-time

The date and time when the key expired.

created_at : string/date-time

The date and time when the key was created.

expired : boolean

True if the key has expired.

constraint_count : integer

The number of constraints on the created key

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.

use_count : integer

The number of times the key has been used.

list_me()

Show info about the logged-in user

Returns email : string

This user's email address.

preferences : dict

This user's preferences.

id : integer

The ID of this user.

feature_flags : dict

The feature flag settings for this user.

initials : string

This user's initials.

username : string

This user's username.

name : string

This user's name.

last_checked_announcements : string/date-time

The date and time at which the user last checked their announcements.

custom_branding : string

The branding of Platform for this user.

groups : list:

An array of `all` the groups this user `is in`.

- `id` : integer

The ID of this group.

- `name` : string

The name of this group.

```
- organization_id : integer
    The organization associated with this group.
```

list_me_api_accesses (**kwargs)

Show API accesses made by or on behalf of the logged-in user

Parameters **limit** : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

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 **api_key_id** : integer

id : integer

created_at : string/time

processing_seconds : number/float

requested_version : integer

status_code : string

method : string

ip : string

path : string

patch_me (**kwargs)

Update info about the logged-in user

Parameters **preferences** : dict, optional:

```
- result_index_order_field : string
    Order field for the results index page.
- script_index_status_filter : string
    Status filter for the scripts index page.
- project_index_archived_filter : string
    Archived filter for the projects index page.
- model_index_order_dir : string
    Order direction for the models index page.
- export_index_type_filter : string
    Type filter for the exports index page.
- script_index_type_filter : string
    Type filter for the scripts index page.
- script_index_author_filter : string
```

```
    Author filter for the scripts index page.
- export_index_status_filter : string
    Status filter for the exports index page.
- export_index_order_field : string
    Order field for the exports index page.
- result_index_author_filter : string
    Author filter for the results index page.
- project_detail_author_filter : string
    Author filter for projects detail pages.
- import_index_status_filter : string
    Status filter for the imports index page.
- model_index_status_filter : string
    Status filter for the models index page.
- model_index_author_filter : string
    Author filter for the models index page.
- model_index_order_field : string
    Order field for the models index page.
- import_index_order_dir : string
    Order direction for the imports index page.
- model_index_archived_filter : string
    Archived filter for the models index page.
- import_index_dest_filter : string
    Destination filter for the imports index page.
- preferred_server_id : integer
    ID of preferred server.
- script_index_archived_filter : string
    Archived filter for the scripts 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.
- project_detail_archived_filter : string
    Archived filter for the projects detail pages.
- script_index_order_field : string
    Order field for the scripts index page.
- project_detail_order_dir : string
    Order direction for projects detail pages.
- import_index_author_filter : string
    Author filter for the imports index page.
- report_index_thumbnail_view : string
    Thumbnail view for the reports index page.
- import_index_order_field : string
    Order field for the imports index page.
- project_detail_type_filter : string
    Type filter for projects detail pages.
- import_index_archived_filter : string
    Archived filter for the imports index page.
- export_index_author_filter : string
    Author filter for the exports index page.
- app_index_order_field : string
    Order field for the apps index pages.
- export_index_order_dir : string
    Order direction for the exports index page.
- project_detail_order_field : string
    Order field for projects detail pages.
- project_index_author_filter : string
    Author filter for the projects index page.
- app_index_order_dir : string
```

```

    Oder direction for the apps index pages.
- result_index_archived_filter : string
    Archived filter for the results index page.
- project_index_order_dir : string
    Order direction for the projects index page.
- script_index_order_dir : string
    Order direction for the scripts 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.
- model_index_thumbnail_view : string
    Thumbnail view for the models index page.

```

last_checked_announcements : string/date-time, optional

The date and time at which the user last checked their announcements.

Returns email : string

This user's email address.

preferences : dict

This user's preferences.

id : integer

The ID of this user.

feature_flags : dict

The feature flag settings for this user.

initials : string

This user's initials.

username : string

This user's username.

name : string

This user's name.

last_checked_announcements : string/date-time

The date and time at which the user last checked their announcements.

custom_branding : string

The branding of Platform for this user.

groups : list:

```

An array of all the groups this user is in.
- id : integer
    The ID of this group.
- name : string
    The name of this group.
- organization_id : integer
    The organization associated with this group.

```

post_api_keys (*id, name, expires_in, **kwargs*)

Create a new API key belonging to the logged-in user

Parameters **id** : string

The ID of the user or 'me'.

name : string

The name of the API key.

expires_in : integer

The number of seconds the key should last for.

constraints : list, optional:

Constraints on the abilities of the created key.

- **get_allowed** : boolean
Whether the constraint allows GET requests.
- **head_allowed** : boolean
Whether the constraint allows HEAD requests.
- **delete_allowed** : boolean
Whether the constraint allows DELETE requests.
- **patch_allowed** : boolean
Whether the constraint allows PATCH requests.
- **post_allowed** : boolean
Whether the constraint allows POST 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).

Returns **scopes** : list

The scopes which the key is permissioned on.

id : integer

The ID of the API key.

expires_at : string/date-time

The date and time when the key expired.

active : boolean

True if the key has neither expired nor been revoked.

expired : boolean

True if the key has expired.

use_count : integer

The number of times the key has been used.

last_used_at : string/date-time

The date and time when the key was last used.

revoked_at : string/date-time

The date and time when the key was revoked.

created_at : string/date-time

The date and time when the key was created.

token : string

The API key.

name : string

The name of the API key.

constraints : list:

```
Constraints on the abilities of the created key
- get_allowed : boolean
    Whether the constraint allows GET requests.
- head_allowed : boolean
    Whether the constraint allows HEAD requests.
- delete_allowed : boolean
    Whether the constraint allows DELETE requests.
- patch_allowed : boolean
    Whether the constraint allows PATCH requests.
- post_allowed : boolean
    Whether the constraint allows POST 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).
```

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 5

Indices and tables

- `genindex`
- `modindex`
- `search`

A

APIClient (class in civis), 20

C

cancel() (civis.polling.PollableResult method), 24

CIVIS_API_KEY, 12, 14–16, 18–21

civis_to_csv() (in module civis.io), 11

civis_to_file() (in module civis.io), 17

Credentials (class in civis.resources._resources), 24

csv_to_civis() (in module civis.io), 12

D

Databases (class in civis.resources._resources), 31

dataframe_to_civis() (in module civis.io), 13

default_credential (civis.APIClient attribute), 21

delete() (civis.resources._resources.Scripts method), 141

delete_api_keys() (civis.resources._resources.Users method), 352

delete_builds() (civis.resources._resources.Models method), 76

delete_containers() (civis.resources._resources.Scripts method), 141

delete_containers_projects() (civis.resources._resources.Scripts method), 141

delete_containers_runs() (civis.resources._resources.Scripts method), 141

delete_containers_shares_groups() (civis.resources._resources.Scripts method), 141

delete_containers_shares_users() (civis.resources._resources.Scripts method), 142

delete_custom() (civis.resources._resources.Scripts method), 142

delete_custom_projects() (civis.resources._resources.Scripts method), 142

delete_custom_runs() (civis.resources._resources.Scripts method), 142

delete_custom_shares_groups() (civis.resources._resources.Scripts method), 142

delete_custom_shares_users() (civis.resources._resources.Scripts method), 143

delete_files_runs() (civis.resources._resources.Imports method), 39

delete_grants() (civis.resources._resources.Reports method), 122

delete_javascript() (civis.resources._resources.Scripts method), 143

delete_javascript_projects() (civis.resources._resources.Scripts method), 143

delete_javascript_runs() (civis.resources._resources.Scripts method), 143

delete_javascript_shares_groups() (civis.resources._resources.Scripts method), 143

delete_javascript_shares_users() (civis.resources._resources.Scripts method), 144

delete_projects() (civis.resources._resources.Files method), 34

delete_projects() (civis.resources._resources.Imports method), 40

delete_projects() (civis.resources._resources.Jobs method), 68

delete_projects() (civis.resources._resources.Models method), 76

delete_projects() (civis.resources._resources.Reports method), 122

delete_python3() (civis.resources._resources.Scripts method), 144

delete_python3_projects() (civis.resources._resources.Scripts method), 144

`delete_python3_runs()` (civis.resources._resources.Scripts method), 144

`delete_python3_shares_groups()` (civis.resources._resources.Scripts method), 144

`delete_python3_shares_users()` (civis.resources._resources.Scripts method), 145

`delete_r()` (civis.resources._resources.Scripts method), 145

`delete_r_projects()` (civis.resources._resources.Scripts method), 145

`delete_r_runs()` (civis.resources._resources.Scripts method), 145

`delete_r_shares_groups()` (civis.resources._resources.Scripts method), 145

`delete_r_shares_users()` (civis.resources._resources.Scripts method), 146

`delete_runs()` (civis.resources._resources.Predictions method), 105

`delete_runs()` (civis.resources._resources.Queries method), 113

`delete_shares_groups()` (civis.resources._resources.Files method), 34

`delete_shares_groups()` (civis.resources._resources.Imports method), 40

`delete_shares_groups()` (civis.resources._resources.Jobs method), 68

`delete_shares_groups()` (civis.resources._resources.Models method), 76

`delete_shares_groups()` (civis.resources._resources.Reports method), 122

`delete_shares_users()` (civis.resources._resources.Files method), 34

`delete_shares_users()` (civis.resources._resources.Imports method), 40

`delete_shares_users()` (civis.resources._resources.Jobs method), 68

`delete_shares_users()` (civis.resources._resources.Models method), 77

`delete_shares_users()` (civis.resources._resources.Reports method), 122

`delete_sql()` (civis.resources._resources.Scripts method), 146

`delete_sql_projects()` (civis.resources._resources.Scripts method), 146

`delete_sql_runs()` (civis.resources._resources.Scripts method), 146

`delete_sql_shares_groups()` (civis.resources._resources.Scripts method), 146

`delete_sql_shares_users()` (civis.resources._resources.Scripts method), 147

`delete_syncs()` (civis.resources._resources.Imports method), 40

`delete_whitelist_ips()` (civis.resources._resources.Databases method), 31

E

environment variable
CIVIS_API_KEY, 12, 14–16, 18–21

F

`failed()` (civis.polling.PollableResult method), 24

`file_to_civis()` (in module civis.io), 18

Files (class in civis.resources._resources), 33

G

`get()` (civis.resources._resources.Credentials method), 26

`get()` (civis.resources._resources.Files method), 34

`get()` (civis.resources._resources.Imports method), 40

`get()` (civis.resources._resources.Jobs method), 68

`get()` (civis.resources._resources.Models method), 77

`get()` (civis.resources._resources.Predictions method), 105

`get()` (civis.resources._resources.Queries method), 114

`get()` (civis.resources._resources.Reports method), 123

`get()` (civis.resources._resources.Scripts method), 147

`get()` (civis.resources._resources.Tables method), 335

`get()` (civis.resources._resources.Users method), 353

`get_api_keys()` (civis.resources._resources.Users method), 355

`get_aws_credential_id()` (civis.APIClient method), 21

`get_builds()` (civis.resources._resources.Models method), 81

`get_containers()` (civis.resources._resources.Scripts method), 150

`get_containers_runs()` (civis.resources._resources.Scripts method), 154

`get_custom()` (civis.resources._resources.Scripts method), 155

`get_custom_runs()` (civis.resources._resources.Scripts method), 158

`get_database_credential_id()` (civis.APIClient method), 22

`get_database_id()` (civis.APIClient method), 22

`get_enhancements_cass_ncoa()` (civis.resources._resources.Tables method), 338

`get_enhancements_geocodings()` (civis.resources._resources.Tables method), 339

`get_enhancements_prepared_matchings()` (civis.resources._resources.Tables method), 339

- [get_enhancements_table_matchings\(\)](#) (civis.resources._resources.Tables method), [340](#)
[get_files_runs\(\)](#) (civis.resources._resources.Imports method), [43](#)
[get_javascript\(\)](#) (civis.resources._resources.Scripts method), [158](#)
[get_javascript_runs\(\)](#) (civis.resources._resources.Scripts method), [161](#)
[get_python3\(\)](#) (civis.resources._resources.Scripts method), [162](#)
[get_python3_runs\(\)](#) (civis.resources._resources.Scripts method), [166](#)
[get_r\(\)](#) (civis.resources._resources.Scripts method), [166](#)
[get_r_runs\(\)](#) (civis.resources._resources.Scripts method), [170](#)
[get_runs\(\)](#) (civis.resources._resources.Predictions method), [107](#)
[get_runs\(\)](#) (civis.resources._resources.Queries method), [115](#)
[get_sql\(\)](#) (civis.resources._resources.Scripts method), [170](#)
[get_sql_runs\(\)](#) (civis.resources._resources.Scripts method), [174](#)
[get_table_id\(\)](#) (civis.APIClient method), [23](#)
[get_whitelist_ips\(\)](#) (civis.resources._resources.Databases method), [31](#)
- I**
- Imports (class in civis.resources._resources), [39](#)
- J**
- Jobs (class in civis.resources._resources), [67](#)
- L**
- [list\(\)](#) (civis.resources._resources.Credentials method), [26](#)
[list\(\)](#) (civis.resources._resources.Databases method), [32](#)
[list\(\)](#) (civis.resources._resources.Imports method), [44](#)
[list\(\)](#) (civis.resources._resources.Jobs method), [69](#)
[list\(\)](#) (civis.resources._resources.Models method), [82](#)
[list\(\)](#) (civis.resources._resources.Predictions method), [108](#)
[list\(\)](#) (civis.resources._resources.Queries method), [115](#)
[list\(\)](#) (civis.resources._resources.Reports method), [124](#)
[list\(\)](#) (civis.resources._resources.Scripts method), [174](#)
[list\(\)](#) (civis.resources._resources.Tables method), [340](#)
[list\(\)](#) (civis.resources._resources.Users method), [356](#)
[list_api_keys\(\)](#) (civis.resources._resources.Users method), [357](#)
[list_builds\(\)](#) (civis.resources._resources.Models method), [85](#)
[list_children\(\)](#) (civis.resources._resources.Jobs method), [70](#)
[list_columns\(\)](#) (civis.resources._resources.Tables method), [342](#)
[list_containers_projects\(\)](#) (civis.resources._resources.Scripts method), [177](#)
[list_containers_runs\(\)](#) (civis.resources._resources.Scripts method), [177](#)
[list_containers_runs_logs\(\)](#) (civis.resources._resources.Scripts method), [178](#)
[list_containers_shares\(\)](#) (civis.resources._resources.Scripts method), [179](#)
[list_custom\(\)](#) (civis.resources._resources.Scripts method), [179](#)
[list_custom_projects\(\)](#) (civis.resources._resources.Scripts method), [181](#)
[list_custom_runs\(\)](#) (civis.resources._resources.Scripts method), [182](#)
[list_custom_runs_logs\(\)](#) (civis.resources._resources.Scripts method), [183](#)
[list_custom_shares\(\)](#) (civis.resources._resources.Scripts method), [183](#)
[list_files_runs\(\)](#) (civis.resources._resources.Imports method), [46](#)
[list_history\(\)](#) (civis.resources._resources.Scripts method), [184](#)
[list_javascript_projects\(\)](#) (civis.resources._resources.Scripts method), [185](#)
[list_javascript_runs\(\)](#) (civis.resources._resources.Scripts method), [186](#)
[list_javascript_runs_logs\(\)](#) (civis.resources._resources.Scripts method), [186](#)
[list_javascript_shares\(\)](#) (civis.resources._resources.Scripts method), [187](#)
[list_me\(\)](#) (civis.resources._resources.Users method), [358](#)
[list_me_api_accesses\(\)](#) (civis.resources._resources.Users method), [359](#)
[list_parents\(\)](#) (civis.resources._resources.Jobs method), [71](#)
[list_projects\(\)](#) (civis.resources._resources.Files method), [35](#)
[list_projects\(\)](#) (civis.resources._resources.Imports method), [47](#)
[list_projects\(\)](#) (civis.resources._resources.Jobs method), [72](#)
[list_projects\(\)](#) (civis.resources._resources.Models method), [87](#)
[list_projects\(\)](#) (civis.resources._resources.Reports method), [126](#)
[list_python3_projects\(\)](#) (civis.resources._resources.Scripts method), [188](#)
[list_python3_runs\(\)](#) (civis.resources._resources.Scripts method), [188](#)

`list_python3_runs_logs()`
(`civis.resources._resources.Scripts` method), 189

`list_python3_shares()` (`civis.resources._resources.Scripts` method), 190

`list_r_projects()` (`civis.resources._resources.Scripts` method), 191

`list_r_runs()` (`civis.resources._resources.Scripts` method), 191

`list_r_runs_logs()` (`civis.resources._resources.Scripts` method), 192

`list_r_shares()` (`civis.resources._resources.Scripts` method), 193

`list_runs()` (`civis.resources._resources.Imports` method), 48

`list_runs()` (`civis.resources._resources.Predictions` method), 108

`list_runs()` (`civis.resources._resources.Queries` method), 117

`list_schedules()` (`civis.resources._resources.Models` method), 87

`list_schedules()` (`civis.resources._resources.Predictions` method), 109

`list_schemas()` (`civis.resources._resources.Databases` method), 32

`list_shares()` (`civis.resources._resources.Files` method), 35

`list_shares()` (`civis.resources._resources.Imports` method), 48

`list_shares()` (`civis.resources._resources.Jobs` method), 73

`list_shares()` (`civis.resources._resources.Models` method), 88

`list_shares()` (`civis.resources._resources.Reports` method), 127

`list_sql_projects()` (`civis.resources._resources.Scripts` method), 193

`list_sql_runs()` (`civis.resources._resources.Scripts` method), 194

`list_sql_runs_logs()` (`civis.resources._resources.Scripts` method), 195

`list_sql_shares()` (`civis.resources._resources.Scripts` method), 196

`list_types()` (`civis.resources._resources.Models` method), 89

`list_types()` (`civis.resources._resources.Scripts` method), 197

`list_whitelist_ips()` (`civis.resources._resources.Databases` method), 32

`patch()` (`civis.resources._resources.Models` method), 89

`patch()` (`civis.resources._resources.Predictions` method), 110

`patch()` (`civis.resources._resources.Reports` method), 128

`patch()` (`civis.resources._resources.Scripts` method), 197

`patch()` (`civis.resources._resources.Tables` method), 344

`patch_containers()` (`civis.resources._resources.Scripts` method), 201

`patch_custom()` (`civis.resources._resources.Scripts` method), 207

`patch_javascript()` (`civis.resources._resources.Scripts` method), 211

`patch_me()` (`civis.resources._resources.Users` method), 359

`patch_python3()` (`civis.resources._resources.Scripts` method), 216

`patch_r()` (`civis.resources._resources.Scripts` method), 222

`patch_sql()` (`civis.resources._resources.Scripts` method), 227

`PollableResult` (class in `civis.polling`), 24

`post()` (`civis.resources._resources.Credentials` method), 27

`post()` (`civis.resources._resources.Files` method), 36

`post()` (`civis.resources._resources.Imports` method), 49

`post()` (`civis.resources._resources.Models` method), 91

`post()` (`civis.resources._resources.Queries` method), 118

`post()` (`civis.resources._resources.Reports` method), 130

`post()` (`civis.resources._resources.Scripts` method), 232

`post()` (`civis.resources._resources.Tables` method), 345

`post_api_keys()` (`civis.resources._resources.Users` method), 361

`post_authenticate()` (`civis.resources._resources.Credentials` method), 28

`post_builds()` (`civis.resources._resources.Models` method), 96

`post_cancel()` (`civis.resources._resources.Scripts` method), 237

`post_containers()` (`civis.resources._resources.Scripts` method), 237

`post_containers_runs()` (`civis.resources._resources.Scripts` method), 243

`post_custom()` (`civis.resources._resources.Scripts` method), 243

`post_custom_runs()` (`civis.resources._resources.Scripts` method), 247

`post_enhancements_cass_ncoa()`
(`civis.resources._resources.Tables` method), 346

`post_enhancements_geocodings()`
(`civis.resources._resources.Tables` method), 347

`post_enhancements_prepared_matchings()`
(`civis.resources._resources.Tables` method),

M

`Models` (class in `civis.resources._resources`), 76

P

`PaginatedResponse` (class in `civis.response`), 23

- 347
- `post_enhancements_table_matchings()`
(`civis.resources._resources.Tables` method), 348
- `post_files()` (`civis.resources._resources.Imports` method), 53
- `post_files_runs()` (`civis.resources._resources.Imports` method), 54
- `post_grants()` (`civis.resources._resources.Reports` method), 132
- `post_javascript()` (`civis.resources._resources.Scripts` method), 248
- `post_javascript_runs()` (`civis.resources._resources.Scripts` method), 253
- `post_python3()` (`civis.resources._resources.Scripts` method), 253
- `post_python3_runs()` (`civis.resources._resources.Scripts` method), 259
- `post_r()` (`civis.resources._resources.Scripts` method), 259
- `post_r_runs()` (`civis.resources._resources.Scripts` method), 265
- `post_refresh()` (`civis.resources._resources.Tables` method), 349
- `post_run()` (`civis.resources._resources.Scripts` method), 265
- `post_runs()` (`civis.resources._resources.Imports` method), 55
- `post_runs()` (`civis.resources._resources.Jobs` method), 73
- `post_runs()` (`civis.resources._resources.Predictions` method), 112
- `post_runs()` (`civis.resources._resources.Queries` method), 119
- `post_sql()` (`civis.resources._resources.Scripts` method), 265
- `post_sql_runs()` (`civis.resources._resources.Scripts` method), 270
- `post_syncs()` (`civis.resources._resources.Imports` method), 55
- `post_temporary()` (`civis.resources._resources.Credentials` method), 29
- `post_trigger_email()` (`civis.resources._resources.Jobs` method), 73
- `post_whitelist_ips()` (`civis.resources._resources.Databases` method), 32
- `Predictions` (class in `civis.resources._resources`), 105
- `put()` (`civis.resources._resources.Credentials` method), 29
- `put()` (`civis.resources._resources.Imports` method), 56
- `put_archive()` (`civis.resources._resources.Imports` method), 61
- `put_archive()` (`civis.resources._resources.Models` method), 97
- `put_archive()` (`civis.resources._resources.Reports` method), 134
- `put_containers()` (`civis.resources._resources.Scripts` method), 271
- `put_containers_archive()`
(`civis.resources._resources.Scripts` method), 277
- `put_containers_projects()`
(`civis.resources._resources.Scripts` method), 281
- `put_containers_shares_groups()`
(`civis.resources._resources.Scripts` method), 281
- `put_containers_shares_users()`
(`civis.resources._resources.Scripts` method), 282
- `put_custom()` (`civis.resources._resources.Scripts` method), 283
- `put_custom_archive()` (`civis.resources._resources.Scripts` method), 287
- `put_custom_projects()` (`civis.resources._resources.Scripts` method), 289
- `put_custom_shares_groups()`
(`civis.resources._resources.Scripts` method), 290
- `put_custom_shares_users()`
(`civis.resources._resources.Scripts` method), 291
- `put_javascript()` (`civis.resources._resources.Scripts` method), 291
- `put_javascript_archive()` (`civis.resources._resources.Scripts` method), 296
- `put_javascript_projects()`
(`civis.resources._resources.Scripts` method), 300
- `put_javascript_shares_groups()`
(`civis.resources._resources.Scripts` method), 300
- `put_javascript_shares_users()`
(`civis.resources._resources.Scripts` method), 301
- `put_predictions()` (`civis.resources._resources.Models` method), 101
- `put_projects()` (`civis.resources._resources.Files` method), 37
- `put_projects()` (`civis.resources._resources.Imports` method), 64
- `put_projects()` (`civis.resources._resources.Jobs` method), 74
- `put_projects()` (`civis.resources._resources.Models` method), 102
- `put_projects()` (`civis.resources._resources.Reports` method), 136
- `put_python3()` (`civis.resources._resources.Scripts` method), 302
- `put_python3_archive()` (`civis.resources._resources.Scripts` method), 307

`put_python3_projects()` (civis.resources._resources.Scripts method), 311

`put_python3_shares_groups()` (civis.resources._resources.Scripts method), 311

`put_python3_shares_users()` (civis.resources._resources.Scripts method), 312

`put_r()` (civis.resources._resources.Scripts method), 313

`put_r_archive()` (civis.resources._resources.Scripts method), 318

`put_r_projects()` (civis.resources._resources.Scripts method), 322

`put_r_shares_groups()` (civis.resources._resources.Scripts method), 322

`put_r_shares_users()` (civis.resources._resources.Scripts method), 323

`put_schedules()` (civis.resources._resources.Models method), 102

`put_schedules()` (civis.resources._resources.Predictions method), 112

`put_scripts()` (civis.resources._resources.Queries method), 120

`put_shares_groups()` (civis.resources._resources.Files method), 37

`put_shares_groups()` (civis.resources._resources.Imports method), 64

`put_shares_groups()` (civis.resources._resources.Jobs method), 74

`put_shares_groups()` (civis.resources._resources.Models method), 103

`put_shares_groups()` (civis.resources._resources.Reports method), 136

`put_shares_users()` (civis.resources._resources.Files method), 38

`put_shares_users()` (civis.resources._resources.Imports method), 65

`put_shares_users()` (civis.resources._resources.Jobs method), 75

`put_shares_users()` (civis.resources._resources.Models method), 104

`put_shares_users()` (civis.resources._resources.Reports method), 137

`put_sql()` (civis.resources._resources.Scripts method), 324

`put_sql_archive()` (civis.resources._resources.Scripts method), 329

`put_sql_projects()` (civis.resources._resources.Scripts method), 332

`put_sql_shares_groups()` (civis.resources._resources.Scripts method), 332

`put_sql_shares_users()` (civis.resources._resources.Scripts method), 333

`put_syncs()` (civis.resources._resources.Imports method),

66

Q

Queries (class in civis.resources._resources), 113

`query_civis()` (in module civis.io), 20

R

`read_civis()` (in module civis.io), 15

`read_civis_sql()` (in module civis.io), 16

Reports (class in civis.resources._resources), 121

Response (class in civis.response), 23

S

Scripts (class in civis.resources._resources), 138

`succeeded()` (civis.polling.PollableResult method), 24

T

Tables (class in civis.resources._resources), 334

`transfer_table()` (in module civis.io), 19

U

username (civis.APIClient attribute), 23

Users (class in civis.resources._resources), 352