Civis Client Documentation

Release 1.0.0

Civis Analytics

Mar 24, 2017

Contents

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 git+git://github.com/civisanalytics/civis-python.git
```

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

```
git clone https://github.com/civisanalytics/civis-client.git
cd civis-client
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*.

CHAPTER 4

Client API Reference

User Guide

Getting Started

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

```
>>> import civis
```

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

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

civis_to_csv(filename, sql, database[, ...])	Export data from Civis to a local CSV file.
csv_to_civis(filename, database, table[, ...])	Upload the contents of a local CSV file to Civis.
dataframe_to_civis(df, database, table[, ...])	Upload a *pandas DataFrame* into a Civis table.
read_civis(table, database[, columns, ...])	Read data from a Civis table.
read_civis_sql(sql, database[, use_pandas, ...])	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'` or `'append'`. 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'` or `'append'`. 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.

civis_to_file(file_id, buf[, api_key])	Download a file from Civis.
file_to_civis(buf, name[, api_key])	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).

`transfer_table`(source_db, dest_db, ...[, ...])	Transfer a table from one location to another.
`query_civis`(sql, database[, api_key, ...])	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 transfered. Optionally, could be the database ID.

source_table : str

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

dest_table : str

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

job_name : str, optional

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

api_key : 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', resources='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 *`Credentials`* endpoint
databases	An instance of the *`Databases`* endpoint
files	An instance of the *`Files`* endpoint
imports	An instance of the *`Imports`* endpoint
jobs	An instance of the *`Jobs`* endpoint
models	An instance of the *`Models`* endpoint
predictions	An instance of the *`Predictions`* endpoint
queries	An instance of the *`Queries`* endpoint
reports	An instance of the *`Reports`* endpoint
scripts	An instance of the *`Scripts`* endpoint
tables	An instance of the *`Tables`* endpoint
users	An instance of the *`Users`* 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

json_data	(dict) This is *json_data* as it is originally returned to the user without the key names being changed. See Notes.
headers	(dict) This is the header for the API call without changing the key names.
calls_remaining	(int) Number of API calls remaining before rate limit is reached.
rate_limit	(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

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

get(*id*)

Get a credential

Parameters **id** : integer

The ID of the credential.

Returns **description** : string

A long description of the credential.

owner : string

The name of the user who this credential belongs to.

username : string

The username for the credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

type : string

The credential's type.

name : string

The name identifying the credential

id : integer

The ID of the credential.

created_at : string/time

The creation time for this credential.

updated_at : string/time

The last modification time for this credential.

remote_host_name : string

The name of the remote host associated with 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 **description** : string

A long description of the credential.

owner : string

The name of the user who this credential belongs to.

username : string

The username for the credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

type : string

The credential's type.

name : string

The name identifying the credential

id : integer

The ID of the credential.

created_at : string/time

The creation time for this credential.

updated_at : string/time

The last modification time for this credential.

remote_host_name : string

The name of the remote host associated with this credential.

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

Create or update a credential

Parameters **username** : string

The username for the credential.

type : string

password : string

The password for the credential.

description : string, optional

A long description of the credential.

name : string, optional

The name identifying the credential.

remote_host : dict, optional:

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

remote_host_id : integer, optional

The ID of the remote host associated with the credential.

Returns **description** : string

A long description of the credential.

owner : string

The name of the user who this credential belongs to.

username : string

The username for the credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

type : string

The credential's type.

name : string

The name identifying the credential

id : integer

The ID of the credential.

created_at : string/time

The creation time for this credential.

updated_at : string/time

The last modification time for this credential.

remote_host_name : string

The name of the remote host associated with this credential.

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

Authenticate against a remote host

Parameters **url** : string

The URL to your host.

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.

Returns **description** : string

A long description of the credential.

owner : string

The name of the user who this credential belongs to.

username : string

The username for the credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

type : string

The credential's type.

name : string

The name identifying the credential

id : integer

The ID of the credential.

created_at : string/time

The creation time for this credential.

updated_at : string/time

The last modification time for this credential.

remote_host_name : string

The name of the remote host associated with 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 **secret_access_key** : string

The secret part of the credential.

session_token : string

The session token identifier.

access_key : string

The identifier of the credential.

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

Update an existing credential

Parameters **id** : integer

The ID of the credential.

username : string

The username for the credential.

type : string

password : string

The password for the credential.

description : string, optional

A long description of the credential.

name : string, optional

The name identifying the credential.

remote_host : dict, optional:

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

remote_host_id : integer, optional

The ID of the remote host associated with the credential.

Returns **description** : string

A long description of the credential.

owner : string

The name of the user who this credential belongs to.

username : string

The username for the credential.

remote_host_id : integer

The ID of the remote host associated with this credential.

type : string

The credential's type.

name : string

The name identifying the credential

id : integer

The ID of the credential.

created_at : string/time

The creation time for this credential.

updated_at : string/time

The last modification time for this credential.

remote_host_name : string

The name of the remote host associated with this credential.

Databases

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

Methods

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

`delete_whitelist_ips` (*id*, *whitelisted_ip_id*)

Remove a whitelisted IP address

Parameters **id** : integer

The ID of the database this rule is applied to.

whitelisted_ip_id : integer

The ID of this whitelisted IP address.

Returns None

Response code 204: success

`get_whitelist_ips` (*id*, *whitelisted_ip_id*)

View details about a whitelisted IP

Parameters **id** : integer

The ID of the database this rule is applied to.

whitelisted_ip_id : integer

The ID of this whitelisted IP address.

Returns **is_active** : boolean

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

authorized_by : string

The user who authorized this rule.

remote_host_id : integer

The ID of the database this rule is applied to.

security_group_id : string

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

id : integer

The ID of this whitelisted IP address.

created_at : string/time

The time this rule was created.

updated_at : string/time

The time this rule was last updated.

subnet_mask : string

The subnet mask that is allowed by this rule.

list()

List databases

Returns **id** : integer

The ID for the database.

name : string

The name of the database.

list_schemas(*id*)

List schemas in this database

Parameters **id** : integer

The ID of the database.

Returns **schema** : string

The name of a schema.

list_whitelist_ips(*id*)

List whitelisted IPs for the specified database

Parameters **id** : integer

The ID for the database.

Returns **remote_host_id** : integer

The ID of the database this rule is applied to.

security_group_id : string

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

id : integer

The ID of this whitelisted IP address.

created_at : string/time

The time this rule was created.

updated_at : string/time

The time this rule was last updated.

subnet_mask : string

The subnet mask that is allowed by this rule.

post_whitelist_ips (*id*, *subnet_mask*)
Whitelist an IP address

Parameters **id** : integer

The ID of the database this rule is applied to.

subnet_mask : string

The subnet mask that is allowed by this rule.

Returns **is_active** : boolean

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

authorized_by : string

The user who authorized this rule.

remote_host_id : integer

The ID of the database this rule is applied to.

security_group_id : string

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

id : integer

The ID of this whitelisted IP address.

created_at : string/time

The time this rule was created.

updated_at : string/time

The time this rule was last updated.

subnet_mask : string

The subnet mask that is allowed by this rule.

Files

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

Methods

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

delete_projects(*id*, *project_id*)

Remove a Data::S3File from a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_shares_groups(*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users(*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get(*id*)

Get details about a file

Parameters **id** : integer

The ID of the file object.

Returns **download_url** : string

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

name : string

The file name.

file_url : string

The URL that may be used to download 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.

file_size : integer

The file size.

created_at : string/date-time

The date and time the file was created.

list_projects(*id*)

List the projects a Data::S3File belongs to

Parameters **id** : integer

The ID of the resource.

Returns **description** : string

A description of the project

created_at : string/time

archived : string

The archival status of the requested object(s).

name : string

The name of this project.

auto_share : boolean

id : integer

The ID for this project.

users : list:

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

updated_at : string/time

author : dict:

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

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

list_shares(*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **writers** : 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.

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
```

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 **upload_fields** : dict

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

name : string

The file name.

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.

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.

file_size : integer

The file size.

created_at : string/date-time

The date and time the file was created.

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 **writers** : 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.

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
```

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

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

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

Returns **writers** : 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.

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
```

Imports

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

Methods

delete_files_runs(id, run_id)	Cancel a run
delete_projects(id, project_id)	Remove a JobTypes::Import from a project
delete_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_shares_users(id, user_id)	Revoke the permissions a user has on this object
delete_syncs(id, sync_id)	Delete a sync
get(id)	Get details about an import
get_files_runs(id, run_id)	Check status of a run
list(**kwargs)	List imports
list_files_runs(id, **kwargs)	List runs for the given import
list_projects(id)	List the projects a JobTypes::Import belongs to
list_runs(id)	Get the run history of this import
list_shares(id)	List users and groups permissioned on this object
post(is_outbound, sync_type, name, **kwargs)	Create a new import configuration
post_files(name, schema, credential_id, ...)	Initate an import of a tabular file into the platform
post_files_runs(id)	Start a run
post_runs(id)	Run an import
post_syncs(id, destination, source, **kwargs)	Create a sync
put(id, is_outbound, sync_type, name, **kwargs)	Update an import
put_archive(id, status)	Update the archive status of this object
put_projects(id, project_id)	Add a JobTypes::Import to a project
put_shares_groups(id, group_ids, ...)	Set the permissions groups has on this object
put_shares_users(id, user_ids, permission_level)	Set the permissions users have on this object
put_syncs(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 **is_outbound** : boolean

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

last_run : dict:

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

sync_type : string

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

notifications : dict:

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

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this import.

created_at : string/date-time

updated_at : string/date-time

archived : string

The archival status of the requested object(s).

state : string

hidden : boolean

The hidden status of the object.

parent_id : integer

Parent id to trigger this import from

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

id : integer

The ID for the import.

syncs : list:

```
List of syncs.
- destination : dict::
    - path : string
        The schema.tablename to sync to.
- id : integer
- source : dict::
    - id : integer
        The ID of the table or file, if available.
    - path : string
        The path of the dataset to sync from; for a database source,
        schema.tablename.
- advanced_options : dict::
    - sortkey2 : string
    - invalid_char_replacement : string
    - soql_query : string
    - contact_lists : string
    - identity_column : string
    - verify_table_row_counts : boolean
    - truncate_long_lines : boolean
    - max_errors : integer
    - first_row_is_header : boolean
    - row_chunk_size : integer
```

```
    - column_delimiter : string
    - existing_table_rows : string
    - last_modified_column : string
    - wipe_destination_table : boolean
    - partition_schema_name : string
    - partition_column_name : string
    - partition_table_partition_column_max_name : string
    - sortkey1 : string
    - export_action : string
    - partition_table_name : string
    - distkey : string
    - partition_table_partition_column_min_name : string
    - mysql_catalog_matches_schema : boolean
```

running_as : dict:

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

schedule : dict:

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

user : dict:

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

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 **finished_at** : string/time

The time the last run completed.

state : string

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

import_id : integer

The ID of the import.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

`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 **is_outbound** : boolean

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

last_run : dict:

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

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.

created_at : string/date-time

updated_at : string/date-time

archived : string

The archival status of the requested object(s).

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

id : integer

The ID for the import.

schedule : dict:

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

user : dict:

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

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 **finished_at** : string/time

The time the last run completed.

state : string

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

import_id : integer

The ID of the import.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

`list_projects`(*id*)

List the projects a JobTypes::Import belongs to

Parameters **id** : integer

The ID of the resource.

Returns **description** : string

A description of the project

created_at : string/time

archived : string

The archival status of the requested object(s).

name : string

The name of this project.

auto_share : boolean

id : integer

The ID for this project.

users : list:

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

updated_at : string/time

author : dict:

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

list_runs(*id*)

Get the run history of this import

Parameters **id** : integer

Returns **finished_at** : string/time

The time that the run completed.

state : string

id : integer

error : string

The error message for this run, if present.

created_at : string/time

The time that the run was queued.

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 **writers** : 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.

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
```

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

Create a new import configuration

Parameters **is_outbound** : boolean

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.

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
```

time_zone : string, optional

The time zone of this import.

parent_id : integer, optional

Parent id to trigger this import from

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
```

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

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

schedule : dict, optional:

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

hidden : boolean, optional

The hidden status of the object.

Returns **is_outbound** : boolean

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

last_run : dict:

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

sync_type : string

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

notifications : dict:

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

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this import.

created_at : string/date-time

updated_at : string/date-time

archived : string

The archival status of the requested object(s).

state : string

hidden : boolean

The hidden status of the object.

parent_id : integer

Parent id to trigger this import from

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

id : integer

The ID for the import.

syncs : list:

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

```
    - partition_table_partition_column_max_name : string
    - sortkey1 : string
    - export_action : string
    - partition_table_name : string
    - distkey : string
    - partition_table_partition_column_min_name : string
    - mysql_catalog_matches_schema : boolean
```

running_as : dict:

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

schedule : dict:

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

user : dict:

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

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

Initate an import of a tabular file into the platform

Parameters **name** : string

The name of the destination table.

schema : string

The schema of the destination table.

credential_id : integer

The id of the credentials to be used when performing the database import.

remote_host_id : integer

The id of the destination database host.

sortkey2 : string, optional

The second column in a compound sortkey for the table.

multipart : boolean, optional

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

first_row_is_header : boolean, optional

A boolean value indicating whether or not the first row is a header row. If first_row_is_header is null or omitted, it will be auto-detected.

sortkey1 : string, optional

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

max_errors : integer, optional

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

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".

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.

Returns **run_uri** : string

The URI to POST to once the file upload is complete. After uploading the file using the URI given in the uploadUri attribute of the reponse, POST to this URI to initiate the import of your uploaded file into the platform.

upload_fields : dict

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

id : integer

The id of the import.

upload_uri : string

The URI which may be used to upload a tabular file for import. You must use this URI to upload the file you wish imported and then inform the Civis API when your upload is complete using the URI given by the runUri field of this reponse.

`post_files_runs`(*id*)

Start a run

Parameters **id** : integer

The ID of the import.

Returns **finished_at** : string/time

The time the last run completed.

state : string

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

import_id : integer

The ID of the import.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

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:

```
- sortkey2 : string
- invalid_char_replacement : string
- soql_query : string
- contact_lists : string
- identity_column : string
- verify_table_row_counts : boolean
- truncate_long_lines : boolean
- max_errors : integer
- first_row_is_header : boolean
```

```
- row_chunk_size : integer
- column_delimiter : string
- existing_table_rows : string
- last_modified_column : string
- wipe_destination_table : boolean
- partition_schema_name : string
- partition_column_name : string
- partition_table_partition_column_max_name : string
- sortkey1 : string
- export_action : string
- partition_table_name : string
- distkey : string
- partition_table_partition_column_min_name : string
- mysql_catalog_matches_schema : boolean
```

Returns **destination** : dict:

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

id : integer

source : dict:

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

advanced_options : dict:

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

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

Update an import

Parameters **id** : integer

The ID for the import.

is_outbound : boolean

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.

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
```

time_zone : string, optional

The time zone of this import.

parent_id : integer, optional

Parent id to trigger this import from

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
```

next_run_at : string/time, optional

The time of the next scheduled run.

notifications : dict, optional:

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

schedule : dict, optional:

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

Returns **is_outbound** : boolean

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

last_run : dict:

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

sync_type : string

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

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
```

```
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this import.

created_at : string/date-time

updated_at : string/date-time

archived : string

The archival status of the requested object(s).

state : string

hidden : boolean

The hidden status of the object.

parent_id : integer

Parent id to trigger this import from

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

id : integer

The ID for the import.

syncs : list:

```
List of syncs.
- destination : dict::
    - path : string
```

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

running_as : dict:

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

schedule : dict:

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

```
    Minutes of the day it is scheduled on
```

user : dict:

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

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 **is_outbound** : boolean

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

last_run : dict:

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

sync_type : string

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

notifications : dict:

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

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this import.

created_at : string/date-time

updated_at : string/date-time

archived : string

The archival status of the requested object(s).

state : string

hidden : boolean

The hidden status of the object.

parent_id : integer

Parent id to trigger this import from

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

id : integer

The ID for the import.

syncs : list:

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

running_as : dict:

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

schedule : dict:

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

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per␣
↪hour
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

user : dict:

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

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 **writers** : 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.

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
```

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

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

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

Returns **writers** : 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.

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
```

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:

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

Returns **destination** : dict:

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

id : integer

source : dict:

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

advanced_options : dict:

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

Jobs

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

Methods

delete_projects(id, project_id)	Remove a Job from a project
delete_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_shares_users(id, user_id)	Revoke the permissions a user has on this object
get(id)	Show basic job info
list(**kwargs)	List jobs
list_children(id)	Show nested tree of children that this job triggers
list_parents(id)	Show chain of parents as a list that this job triggers from
list_projects(id)	List the projects a Job belongs to

Continued on next page

Table 4.8 – continued from previous page

`list_shares`(id)	List users and groups permissioned on this object
`post_runs`(id)	Run a job
`post_trigger_email`(id)	Generate and retrieve trigger email address
`put_projects`(id, project_id)	Add a Job to a project
`put_shares_groups`(id, group_ids, ...)	Set the permissions groups has on this object
`put_shares_users`(id, user_ids, permission_level)	Set the permissions users have on this object

delete_projects(*id*, *project_id*)

Remove a Job from a project

Parameters **id** : integer

ID of the resource

project_id : integer

The ID of the project

Returns None

Response code 204: success

delete_shares_groups(*id*, *group_id*)

Revoke the permissions a group has on this object

Parameters **id** : integer

ID of the resource to be revoked

group_id : integer

ID of the group

Returns None

Response code 204: success

delete_shares_users(*id*, *user_id*)

Revoke the permissions a user has on this object

Parameters **id** : integer

ID of the resource to be revoked

user_id : integer

ID of the user

Returns None

Response code 204: success

get(*id*)

Show basic job info

Parameters **id** : integer

The ID for this job.

Returns **archived** : string

The archival status of the requested object(s).

state : string

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

runs : list:

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

type : string

name : string

last_run : dict:

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

id : integer

hidden : boolean

The hidden status of the object.

created_at : string/date-time

updated_at : string/date-time

list(***kwargs*)

List jobs

Parameters **limit** : integer, optional

The maximum number of jobs to return.

state : string, optional

The job's state. One or more of queued, running, succeeded, failed, and cancelled. Specify multiple values as a comma-separated list (e.g., "A,B").

type : string, optional

The job's type. Specify multiple values as a comma-separated list (e.g., "A,B").

q : string, optional

Query string to search on the id, name, and job type

permission : string, optional

A permissions string, one of "read", "write", or "manage". Lists only jobs for which the current user has that permission.

archived : string, optional

The archival status of the requested object(s).

Returns **archived** : string

The archival status of the requested object(s).

state : string

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

last_run : dict:

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

type : string

name : string

id : integer

created_at : string/date-time

updated_at : string/date-time

list_children(*id*)

Show nested tree of children that this job triggers

Parameters **id** : integer

The ID for this job.

Returns **state** : string

runs : list:

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

type : string

name : string

last_run : dict:

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

id : integer

created_at : string/date-time

updated_at : string/date-time

children : list

list_parents(*id*)

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

Parameters **id** : integer

The ID for this job.

Returns **archived** : string

The archival status of the requested object(s).

state : string

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

runs : list:

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

type : string

name : string

last_run : dict:

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

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

id : integer

hidden : boolean

The hidden status of the object.

created_at : string/date-time

updated_at : string/date-time

list_projects(*id*)

List the projects a Job belongs to

Parameters **id** : integer

The ID of the resource.

Returns **description** : string

A description of the project

created_at : string/time

archived : string

The archival status of the requested object(s).

name : string

The name of this project.

auto_share : boolean

id : integer

The ID for this project.

users : list:

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

updated_at : string/time

author : dict:

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

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

list_shares(*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **writers** : 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.

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
```

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 **writers** : 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.

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
```

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

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

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

Returns **writers** : 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.

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
```

Models

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

Methods

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

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

builds : list:

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

```
    The ID of the model build.
- created_at : string
    The time the model build was created.
- r_squared_error : number/float
    A key metric for continuous models. Nil for other model types.
```

time_zone : string

The time zone of this model.

created_at : string/date-time

The time the model was created.

updated_at : string/date-time

The time the model was updated.

database_id : integer

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

limiting_sql : string

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

credential_id : integer

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

parent_id : integer

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

user : dict:

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

archived : string

The archival status of the requested object(s).

running_as : dict:

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

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

excluded_columns : list

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

description : string

A description of the model.

box_cox_transformation : boolean

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

table_name : string

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

interaction_terms : boolean

Whether to search for interaction terms.

model_name : string

The name of the model.

notifications : dict:

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

hidden : boolean

The hidden status of the object.

primary_key : string

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

cross_validation_parameters : dict

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

number_of_folds : integer

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

predictions : list:

```
The tables upon which the model will be applied.
- limiting_sql : string
    A SQL WHERE clause used to scope the rows to be predicted.
- 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.
- id : integer
    The ID of the model to which to apply the prediction.
- primary_key : list
    The primary key or composite keys of the table being predicted.
- output_table : string
    The qualified name of the table to be created which will␣
↪contain the
    model's predictions.
- schedule : dict::
    - scheduled_days : list
        Day based on numeric value starting at 0 for Sunday
    - scheduled_hours : list
        Hours of the day it is scheduled on
    - scheduled : boolean
        If the object is scheduled
    - scheduled_runs_per_hour : integer
        Alternative to scheduled minutes, number of times to run␣
↪per hour
    - scheduled_minutes : list
        Minutes of the day it is scheduled on
```

current_build_exception : string

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

last_output_location : string

The output JSON for the last build.

active_build_id : integer

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

dependent_variable : string

The dependent variable of the training dataset.

id : integer

The ID of the model.

model_type_id : integer

The ID of the model's type.

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.

schedule : dict:

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

dependent_variable_order : list

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

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 **root_mean_squared_error** : number/float

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

state : string

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

output : string

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

created_at : string

The time the model build was created.

description : string

A description of the model build.

roc_auc : number/float

A key metric for binary, multinomial, and ordinal 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.

name : string

The name of the model build.

transformation_metadata : string

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

id : integer

The ID of the model build.

error : string

The error, if any, returned by the build.

r_squared_error : number/float

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

list(***kwargs*)

List models

Parameters **model_name** : string, optional

If specified, will be used to filter the models returned. Substring matching is supported. (e.g., "modelName=model" will return both "model1" and "my model").

training_table_name : string, optional

If specified, will be used to filter the models returned by the training dataset table name. Substring matching is supported. (e.g., "trainingTableName=table" will return both "table1" and "my_table").

dependent_variable : string, optional

If specified, will be used to filter the models returned by the dependent variable column name. Substring matching is supported. (e.g., "dependentVariable=predictor" will return both "predictor" and "my predictor").

author : string, optional

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

status : string, optional

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

archived : string, optional

The archival status of the requested object(s).

limit : integer, optional

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

page_num : integer, optional

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

order : string, optional

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

order_dir : string, optional

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

iterator : bool, optional

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

Returns **last_run** : dict:

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

builds : list:

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

time_zone : string

The time zone of this model.

created_at : string/date-time

The time the model was created.

updated_at : string/date-time

The time the model was updated.

database_id : integer

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

limiting_sql : string

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

credential_id : integer

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

parent_id : integer

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

model_type_id : integer

The ID of the model's type.

archived : string

The archival status of the requested object(s).

excluded_columns : list

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

description : string

A description of the model.

box_cox_transformation : boolean

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

table_name : string

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

interaction_terms : boolean

Whether to search for interaction terms.

model_name : string

The name of the model.

last_output_location : string

The output JSON for the last build.

primary_key : string

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

cross_validation_parameters : dict

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

number_of_folds : integer

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

predictions : list:

```
The tables upon which the model will be applied.
- limiting_sql : string
    A SQL WHERE clause used to scope the rows to be predicted.
- 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.
- id : integer
```

```
    The ID of the model to which to apply the prediction.
- primary_key : list
    The primary key or composite keys of the table being predicted.
- output_table : string
    The qualified name of the table to be created which will␣
↪contain the
    model's predictions.
```

current_build_exception : string

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

dependent_variable : string

The dependent variable of the training dataset.

id : integer

The ID of the model.

user : dict:

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

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.

schedule : dict:

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

dependent_variable_order : list

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

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 **root_mean_squared_error** : number/float

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

state : string

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

output : string

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

created_at : string

The time the model build was created.

description : string

A description of the model build.

roc_auc : number/float

A key metric for binary, multinomial, and ordinal 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.

name : string

The name of the model build.

transformation_metadata : string

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

id : integer

The ID of the model build.

error : string

The error, if any, returned by the build.

r_squared_error : number/float

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

list_projects(*id*)

List the projects a models belongs to

Parameters **id** : integer

The ID of the resource.

Returns **description** : string

A description of the project

created_at : string/time

archived : string

The archival status of the requested object(s).

name : string

The name of this project.

auto_share : boolean

id : integer

The ID for this project.

users : list:

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

updated_at : string/time

author : dict:

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

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

list_shares(*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **writers** : 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.

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
```

list_types()

List all available model types

Returns **id** : integer

The ID of the model type.

dv_type : string

The type of dependent variable predicted by the model.

algorithm : string

The name of the algorithm used to train the model.

fint_allowed : boolean

Whether this model type supports searching for interaction terms.

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

Update model configuration

Parameters **id** : integer

The ID of the model.

number_of_folds : integer, optional

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

box_cox_transformation : boolean, optional

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

table_name : string, optional

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

interaction_terms : boolean, optional

Whether to search for interaction terms.

model_name : string, optional

The name of the model.

notifications : dict, optional:

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

excluded_columns : list, optional

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

time_zone : string, optional

The time zone of this model.

limiting_sql : string, optional

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

database_id : integer, optional

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

description : string, optional

A description of the model.

primary_key : string, optional

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

credential_id : integer, optional

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

parent_id : integer, optional

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

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]}.

active_build_id : integer, optional

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

dependent_variable : string, optional

The dependent variable of the training dataset.

dependent_variable_order : list, optional

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

schedule : dict, optional:

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

```
    Alternative to scheduled minutes, number of times to run per␣
↪hour
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

model_type_id : integer, optional

The ID of the model's type.

Returns None

Response code 204: success

post (***kwargs*)

Create new configuration for a model

Parameters **number_of_folds** : integer, optional

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

box_cox_transformation : boolean, optional

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

table_name : string, optional

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

interaction_terms : boolean, optional

Whether to search for interaction terms.

model_name : string, optional

The name of the model.

notifications : dict, optional:

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

excluded_columns : list, optional

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

time_zone : string, optional

The time zone of this model.

limiting_sql : string, optional

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

database_id : integer, optional

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

description : string, optional

A description of the model.

primary_key : string, optional

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

credential_id : integer, optional

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

hidden : boolean, optional

The hidden status of the object.

parent_id : integer, optional

The ID of the parent job that will trigger this 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]}.

active_build_id : integer, optional

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

dependent_variable : string, optional

The dependent variable of the training dataset.

dependent_variable_order : list, optional

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

schedule : dict, optional:

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

model_type_id : integer, optional

The ID of the model's type.

Returns **last_run** : dict:

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

builds : list:

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

time_zone : string

The time zone of this model.

created_at : string/date-time

The time the model was created.

updated_at : string/date-time

The time the model was updated.

database_id : integer

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

limiting_sql : string

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

credential_id : integer

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

parent_id : integer

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

user : dict:

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

archived : string

The archival status of the requested object(s).

running_as : dict:

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

excluded_columns : list

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

description : string

A description of the model.

box_cox_transformation : boolean

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

table_name : string

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

interaction_terms : boolean

Whether to search for interaction terms.

model_name : string

The name of the model.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
```

```
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

hidden : boolean

The hidden status of the object.

primary_key : string

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

cross_validation_parameters : dict

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

number_of_folds : integer

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

predictions : list:

```
The tables upon which the model will be applied.
- limiting_sql : string
    A SQL WHERE clause used to scope the rows to be predicted.
- 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.
- id : integer
    The ID of the model to which to apply the prediction.
- primary_key : list
    The primary key or composite keys of the table being predicted.
- output_table : string
    The qualified name of the table to be created which will␣
↪contain the
    model's predictions.
- schedule : dict::
    - scheduled_days : list
        Day based on numeric value starting at 0 for Sunday
    - scheduled_hours : list
        Hours of the day it is scheduled on
    - scheduled : boolean
        If the object is scheduled
    - scheduled_runs_per_hour : integer
        Alternative to scheduled minutes, number of times to run␣
↪per hour
    - scheduled_minutes : list
```

```
        Minutes of the day it is scheduled on
```

current_build_exception : string

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

last_output_location : string

The output JSON for the last build.

active_build_id : integer

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

dependent_variable : string

The dependent variable of the training dataset.

id : integer

The ID of the model.

model_type_id : integer

The ID of the model's type.

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.

schedule : dict:

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

dependent_variable_order : list

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

post_builds(*id*)

Start a build

Parameters **id** : integer

The ID of the model.

Returns **root_mean_squared_error** : number/float

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

state : string

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

output : string

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

created_at : string

The time the model build was created.

description : string

A description of the model build.

roc_auc : number/float

A key metric for binary, multinomial, and ordinal 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.

name : string

The name of the model build.

transformation_metadata : string

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

id : integer

The ID of the model build.

error : string

The error, if any, returned by the build.

r_squared_error : number/float

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

put_archive(*id*, *status*)

Update the archive status of this object

Parameters **id** : integer

The ID of the object.

status : boolean

The desired archived status of the object.

Returns **last_run** : dict:

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

builds : list:

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

time_zone : string

The time zone of this model.

created_at : string/date-time

The time the model was created.

updated_at : string/date-time

The time the model was updated.

database_id : integer

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

limiting_sql : string

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

credential_id : integer

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

parent_id : integer

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

user : dict:

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

archived : string

The archival status of the requested object(s).

running_as : dict:

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

excluded_columns : list

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

description : string

A description of the model.

box_cox_transformation : boolean

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

table_name : string

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

interaction_terms : boolean

Whether to search for interaction terms.

model_name : string

The name of the model.

notifications : dict:

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

hidden : boolean

The hidden status of the object.

primary_key : string

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

cross_validation_parameters : dict

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

number_of_folds : integer

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

predictions : list:

```
The tables upon which the model will be applied.
- limiting_sql : string
    A SQL WHERE clause used to scope the rows to be predicted.
- 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.
- id : integer
    The ID of the model to which to apply the prediction.
- primary_key : list
    The primary key or composite keys of the table being predicted.
- output_table : string
    The qualified name of the table to be created which will␣
↪contain the
    model's predictions.
- schedule : dict::
    - scheduled_days : list
        Day based on numeric value starting at 0 for Sunday
    - scheduled_hours : list
        Hours of the day it is scheduled on
    - scheduled : boolean
        If the object is scheduled
    - scheduled_runs_per_hour : integer
        Alternative to scheduled minutes, number of times to run␣
↪per hour
    - scheduled_minutes : list
        Minutes of the day it is scheduled on
```

current_build_exception : string

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

last_output_location : string

The output JSON for the last build.

active_build_id : integer

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

dependent_variable : string

The dependent variable of the training dataset.

id : integer

The ID of the model.

model_type_id : integer

The ID of the model's type.

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.

schedule : dict:

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

dependent_variable_order : list

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

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

Add a table on which to apply the predictive model

Parameters **id** : integer

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

primary_key : list

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

table_name : string

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

limiting_sql : string, optional

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

output_table : string, optional

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

schedule : dict, optional:

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

Returns **limiting_sql** : string

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

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.

id : integer

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

primary_key : list

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

output_table : string

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

schedule : dict:

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

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_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
```

```
    Hours of the day it is scheduled on
- scheduled : boolean
    If the object is scheduled
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per␣
↪hour
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

Returns **id** : integer

The ID of the model associated with this schedule.

schedule : dict:

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

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 **writers** : 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.

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
```

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

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

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

Returns **writers** : 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.

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
```

Predictions

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

Methods

`delete_runs`(id, run_id)	Cancel a run
`get`(id)	Show the specified prediction
`get_runs`(id, run_id)	Check status of a run
`list`(**kwargs)	List predictions
`list_runs`(id, **kwargs)	List runs for the given prediction
`list_schedules`(id)	Show the prediction schedule
`patch`(id, **kwargs)	Update a prediction
`post_runs`(id)	Start a run
`put_schedules`(id, **kwargs)	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 **finished_at** : string/date-time

The end time of the last run of this prediction.

state : string

The state of the last run of this prediction.

last_run : dict:

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

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

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.

primary_key : list

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

limiting_sql : string

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

model_id : integer

The ID of the model used for this prediction.

scored_table_name : string

The name of the source table for this prediction.

id : integer

The ID of the prediction.

scored_table_id : integer

The ID of the source table for this prediction.

scored_tables : list:

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

```
- schema : string
    The schema of table with created predictions.
- name : string
    The name of table with created predictions.
```

schedule : dict:

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

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 **state** : string

The state of the prediction run.

score_stats : list:

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

exception : string

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

name : string

The name of table created by this predictions run.

id : integer

The ID of the prediction run.

prediction_id : integer

The ID of the prediction.

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 **last_run** : dict:

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

finished_at : string/date-time

The end time of the last run of this prediction.

state : string

The state of the last run of this prediction.

model_id : integer

The ID of the model used for 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.

id : integer

The ID of the 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.

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 **state** : string

The state of the prediction run.

score_stats : list:

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

exception : string

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

name : string

The name of table created by this predictions run.

id : integer

The ID of the prediction run.

prediction_id : integer

The ID of the prediction.

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

score_on_model_build : boolean

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

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

Update a prediction

Parameters **id** : integer

The ID of the prediction.

output_table_name : string, optional

The name of the output table for this prediction.

limiting_sql : string, optional

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

primary_key : list, optional

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

Returns **finished_at** : string/date-time

The end time of the last run of this prediction.

state : string

The state of the last run of this prediction.

last_run : dict:

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

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.

primary_key : list

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

limiting_sql : string

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

model_id : integer

The ID of the model used for this prediction.

scored_table_name : string

The name of the source table for this prediction.

id : integer

The ID of the prediction.

scored_table_id : integer

The ID of the source table for this prediction.

scored_tables : list:

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

schedule : dict:

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

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

post_runs(*id*)

Start a run

Parameters **id** : integer

The ID of the prediction.

Returns **state** : string

The state of the prediction run.

score_stats : list:

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

exception : string

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

name : string

The name of table created by this predictions run.

id : integer

The ID of the prediction run.

prediction_id : integer

The ID of the prediction.

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

```
    Alternative to scheduled minutes, number of times to run per␣
↪hour
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

score_on_model_build : boolean, optional

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

Returns id : integer

ID of the prediction associated with this schedule.

schedule : dict:

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

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

delete_runs(id, run_id)	Cancel a run
get(id)	Get details about a query
get_runs(id, run_id)	Check status of a run
list(**kwargs)	List all queries
list_runs(id, **kwargs)	List runs for the given query
post(preview_rows, database, sql, **kwargs)	Execute a query
post_runs(id)	Start a run
put_scripts(id, script_id)	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 **started_at** : string/date-time

The start time of the last run.

finished_at : string/date-time

The end time of the last run.

script_id : integer

The ID of the script associated with this query.

name : string

The name of the query.

author : dict:

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

credential : integer

The credential ID.

hidden : boolean

The hidden status of the object.

created_at : string/time

updated_at : string/time

result_columns : list

A preview of columns returned by the query.

report_id : integer

The ID of the report associated with this query.

exception : string

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

sql : string

The SQL to execute.

last_run_id : integer

The ID of the last run.

id : integer

The query ID.

database : integer

The database ID.

result_rows : list

A preview of rows returned by the query.

state : string

The state of the last run.

get_runs(*id*, *run_id*)

Check status of a run

Parameters **id** : integer

The ID of the query.

run_id : integer

The ID of the run.

Returns **finished_at** : string/time

The time the last run completed.

state : string

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

query_id : integer

The ID of the query.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

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 **started_at** : string/date-time

The start time of the last run.

finished_at : string/date-time

The end time of the last run.

script_id : integer

The ID of the script associated with this query.

credential : integer

The credential ID.

created_at : string/time

updated_at : string/time

result_columns : list

A preview of columns returned by the query.

report_id : integer

The ID of the report associated with this query.

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.

sql : string

The SQL to execute.

last_run_id : integer

The ID of the last run.

id : integer

The query ID.

database : integer

The database ID.

result_rows : list

A preview of rows returned by the query.

state : string

The state of the last run.

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

List runs for the given query

Parameters **id** : integer

The ID of the query.

limit : integer, optional

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

page_num : integer, optional

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

order : string, optional

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

order_dir : string, optional

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

iterator : bool, optional

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

Returns **finished_at** : string/time

The time the last run completed.

state : string

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

query_id : integer

The ID of the query.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

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

Execute a query

Parameters **preview_rows** : integer

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

database : integer

The database ID.

sql : string

The SQL to execute.

compression : string, optional

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

include_header : boolean, optional

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

unquoted : boolean, optional

If true, will not quote fields.

credential : integer, optional

The credential ID.

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].

filename_prefix : string, optional

The output filename prefix.

interactive : boolean, optional

Deprecated and not used.

Returns **credential** : integer

The credential ID.

script_id : integer

The ID of the script associated with this query.

created_at : string/time

updated_at : string/time

interactive : boolean

Deprecated and not used.

unquoted : boolean

If true, will not quote fields.

column_delimiter : string

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

result_columns : list

A preview of columns returned by the query.

filename_prefix : string

The output filename prefix.

state : string

The state of the last run.

finished_at : string/date-time

The end time of the last run.

last_run_id : integer

The ID of the last run.

compression : string

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

preview_rows : integer

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

hidden : boolean

The hidden status of the object.

started_at : string/date-time

The start time of the last run.

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].

exception : string

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

sql : string

The SQL to execute.

id : integer

The query ID.

database : integer

The database ID.

result_rows : list

A preview of rows returned by the query.

post_runs(*id*)

Start a run

Parameters **id** : integer

The ID of the query.

Returns **finished_at** : string/time

The time the last run completed.

state : string

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

query_id : integer

The ID of the query.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

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 **started_at** : string/date-time

The start time of the last run.

finished_at : string/date-time

The end time of the last run.

script_id : integer

The ID of the script associated with this query.

name : string

The name of the query.

author : dict:

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

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

credential : integer

The credential ID.

hidden : boolean

The hidden status of the object.

created_at : string/time

updated_at : string/time

result_columns : list

A preview of columns returned by the query.

report_id : integer

The ID of the report associated with this query.

exception : string

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

sql : string

The SQL to execute.

last_run_id : integer

The ID of the last run.

id : integer

The query ID.

database : integer

The database ID.

result_rows : list

A preview of rows returned by the query.

state : string

The state of the last run.

Reports

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

Methods

delete_grants(id)	Revoke permisstion for this report to perform Civis platform API operations on
delete_projects(id, project_id)	Remove a Report from a project
delete_shares_groups(id, group_id)	Revoke the permissions a group has on this object

Continued on next page

Table 4.12 – continued from previous page

delete_shares_users(id, user_id)	Revoke the permissions a user has on this object
get(id)	Show a single report
list(**kwargs)	List the reports visible to the current user
list_projects(id)	List the projects a Report belongs to
list_shares(id)	List users and groups permissioned on this object
patch(id, **kwargs)	Update a report
post(**kwargs)	Create a report
post_grants(id)	Grant this report the ability to perform Civis platform API operations on your
put_archive(id, status)	Update the archive status of this object
put_projects(id, project_id)	Add a Report to a project
put_shares_groups(id, group_ids, ...)	Set the permissions groups has on this object
put_shares_users(id, user_ids, permission_level)	Set the permissions users have on this object

delete_grants(*id*)

Revoke permisstion 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 **last_run** : dict:

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

valid_output_file : boolean

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

tableau_id : integer

created_at : string/time

updated_at : string/time

script : dict:

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

archived : string

The archival status of the requested object(s).

name : string

The name of the report.

app_state : dict

Any application state blob for this report.

auth_code_url : string

viz_updated_at : string/time

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

user : dict:

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

finished_at : string/time

The time that the report's last run finished.

state : string

The status of the report's last run.

auth_thumbnail_url : string

URL for a thumbnail of the report.

projects : list:

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

hidden : boolean

The hidden status of the object.

config : string

Any configuration metadata for this report.

template_id : integer

The ID of the template used for this report.

provide_api_key : boolean

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

auth_data_url : string

id : integer

The ID of this report.

api_key : string

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

api_key_id : integer

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

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 **finished_at** : string/time

The time that the report's last run finished.

state : string

The status of the report's last run.

auth_thumbnail_url : string

URL for a thumbnail of the report.

name : string

The name of the report.

tableau_id : integer

created_at : string/time

updated_at : string/time

script : dict:

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

```
- name : string
    The name of the script.
```

template_id : integer

The ID of the template used for this report.

archived : string

The archival status of the requested object(s).

projects : list:

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

viz_updated_at : string/time

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

last_run : dict:

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

id : integer

The ID of this report.

user : dict:

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

list_projects(*id*)

List the projects a Report belongs to

Parameters **id** : integer

The ID of the resource.

Returns **description** : string

A description of the project

created_at : string/time

archived : string

The archival status of the requested object(s).

name : string

The name of this project.

auto_share : boolean

id : integer

The ID for this project.

users : list:

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

updated_at : string/time

author : dict:

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

list_shares(*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **writers** : 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.

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
```

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.

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.

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

Returns last_run : dict:

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

valid_output_file : boolean

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

tableau_id : integer

created_at : string/time

updated_at : string/time

script : dict:

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

archived : string

The archival status of the requested object(s).

name : string

The name of the report.

app_state : dict

Any application state blob for this report.

auth_code_url : string

viz_updated_at : string/time

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

user : dict:

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

finished_at : string/time

The time that the report's last run finished.

state : string

The status of the report's last run.

auth_thumbnail_url : string

URL for a thumbnail of the report.

projects : list:

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

hidden : boolean

The hidden status of the object.

config : string

Any configuration metadata for this report.

template_id : integer

The ID of the template used for this report.

provide_api_key : boolean

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

auth_data_url : string

id : integer

The ID of this report.

api_key : string

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

api_key_id : integer

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

post (***kwargs*)

Create a report

Parameters **template_id** : integer, optional

The ID of the template used for 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.

name : string, optional

The name of the 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.

hidden : boolean, optional

The hidden status of the object.

Returns **last_run** : dict:

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

valid_output_file : boolean

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

tableau_id : integer

created_at : string/time

updated_at : string/time

script : dict:

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

archived : string

The archival status of the requested object(s).

name : string

The name of the report.

app_state : dict

Any application state blob for this report.

auth_code_url : string

viz_updated_at : string/time

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

user : dict:

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

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

finished_at : string/time

The time that the report's last run finished.

state : string

The status of the report's last run.

auth_thumbnail_url : string

URL for a thumbnail of the report.

projects : list:

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

hidden : boolean

The hidden status of the object.

config : string

Any configuration metadata for this report.

template_id : integer

The ID of the template used for this report.

provide_api_key : boolean

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

auth_data_url : string

id : integer

The ID of this report.

api_key : string

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

api_key_id : integer

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

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 **last_run** : dict:

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

valid_output_file : boolean

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

tableau_id : integer

created_at : string/time

updated_at : string/time

script : dict:

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

archived : string

The archival status of the requested object(s).

name : string

The name of the report.

app_state : dict

Any application state blob for this report.

auth_code_url : string

viz_updated_at : string/time

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

user : dict:

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

finished_at : string/time

The time that the report's last run finished.

state : string

The status of the report's last run.

auth_thumbnail_url : string

URL for a thumbnail of the report.

projects : list:

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

hidden : boolean

The hidden status of the object.

config : string

Any configuration metadata for this report.

template_id : integer

The ID of the template used for this report.

provide_api_key : boolean

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

auth_data_url : string

id : integer

The ID of this report.

api_key : string

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

api_key_id : integer

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

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 **last_run** : dict:

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

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

valid_output_file : boolean

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

tableau_id : integer

created_at : string/time

updated_at : string/time

script : dict:

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

archived : string

The archival status of the requested object(s).

name : string

The name of the report.

app_state : dict

Any application state blob for this report.

auth_code_url : string

viz_updated_at : string/time

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

user : dict:

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

finished_at : string/time

The time that the report's last run finished.

state : string

The status of the report's last run.

auth_thumbnail_url : string

URL for a thumbnail of the report.

projects : list:

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

hidden : boolean

The hidden status of the object.

config : string

Any configuration metadata for this report.

template_id : integer

The ID of the template used for this report.

provide_api_key : boolean

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

auth_data_url : string

id : integer

The ID of this report.

api_key : string

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

api_key_id : integer

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

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 **writers** : 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.

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
```

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

Set the permissions users have on this object

Parameters id : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

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

Returns writers : 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.

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
```

Scripts

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

Methods

delete(id)	Delete a script
delete_containers(id)	Delete a container
delete_containers_projects(id, project_id)	Remove a container docker from a project
delete_containers_runs(id, run_id)	Cancel a run
delete_containers_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_containers_shares_users(id, user_id)	Revoke the permissions a user has on this object
delete_custom(id)	Delete a CustomScript
delete_custom_projects(id, project_id)	Remove a Job from a project
delete_custom_runs(id, run_id)	Cancel a run
delete_custom_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_custom_shares_users(id, user_id)	Revoke the permissions a user has on this object
delete_javascript(id)	Delete a JavaScript Script
delete_javascript_projects(id, project_id)	Remove a scripted sql from a project
delete_javascript_runs(id, run_id)	Cancel a run
delete_javascript_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_javascript_shares_users(id, user_id)	Revoke the permissions a user has on this object

Continued on next page

Table 4.13 – continued from previous page

delete_python3(id)	Delete a Python Script
delete_python3_projects(id, project_id)	Remove a python docker from a project
delete_python3_runs(id, run_id)	Cancel a run
delete_python3_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_python3_shares_users(id, user_id)	Revoke the permissions a user has on this object
delete_r(id)	Delete an R Script
delete_r_projects(id, project_id)	Remove a r docker from a project
delete_r_runs(id, run_id)	Cancel a run
delete_r_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_r_shares_users(id, user_id)	Revoke the permissions a user has on this object
delete_sql(id)	Delete a SQL script
delete_sql_projects(id, project_id)	Remove a scripts from a project
delete_sql_runs(id, run_id)	Cancel a run
delete_sql_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_sql_shares_users(id, user_id)	Revoke the permissions a user has on this object
get(id)	Get details about a script
get_containers(id)	View a container
get_containers_runs(id, run_id)	Check status of a run
get_custom(id)	Get a CustomScript
get_custom_runs(id, run_id)	Check status of a run
get_javascript(id)	Get a JavaScript Script
get_javascript_runs(id, run_id)	Check status of a run
get_python3(id)	Get a Python Script
get_python3_runs(id, run_id)	Check status of a run
get_r(id)	Get an R Script
get_r_runs(id, run_id)	Check status of a run
get_sql(id)	Get a SQL script
get_sql_runs(id, run_id)	Check status of a run
list(**kwargs)	List scripts
list_containers_projects(id)	List the projects a container docker belongs to
list_containers_runs(id, **kwargs)	List runs for the given container
list_containers_runs_logs(id, run_id, **kwargs)	Get the logs for a run
list_containers_shares(id)	List users and groups permissioned on this object
list_custom(**kwargs)	List Custom Scripts
list_custom_projects(id)	List the projects a Job belongs to
list_custom_runs(id, **kwargs)	List runs for the given custom
list_custom_runs_logs(id, run_id, **kwargs)	Get the logs for a run
list_custom_shares(id)	List users and groups permissioned on this object
list_history(id)	Get the run history and outputs of this script
list_javascript_projects(id)	List the projects a scripted sql belongs to
list_javascript_runs(id, **kwargs)	List runs for the given javascript
list_javascript_runs_logs(id, run_id, **kwargs)	Get the logs for a run
list_javascript_shares(id)	List users and groups permissioned on this object
list_python3_projects(id)	List the projects a python docker belongs to
list_python3_runs(id, **kwargs)	List runs for the given python
list_python3_runs_logs(id, run_id, **kwargs)	Get the logs for a run
list_python3_shares(id)	List users and groups permissioned on this object

Continued on next page

Table 4.13 – continued from previous page

list_r_projects(id)	List the projects a r docker belongs to
list_r_runs(id, **kwargs)	List runs for the given r
list_r_runs_logs(id, run_id, **kwargs)	Get the logs for a run
list_r_shares(id)	List users and groups permissioned on this object
list_sql_projects(id)	List the projects a scripts belongs to
list_sql_runs(id, **kwargs)	List runs for the given sql
list_sql_runs_logs(id, run_id, **kwargs)	Get the logs for a run
list_sql_shares(id)	List users and groups permissioned on this object
list_types()	List available script types
patch(id, **kwargs)	Update a script
patch_containers(id, **kwargs)	Update a container
patch_custom(id, **kwargs)	Update some attributes of this CustomScript
patch_javascript(id, **kwargs)	Update some attributes of this JavaScript Script
patch_python3(id, **kwargs)	Update some attributes of this Python Script
patch_r(id, **kwargs)	Update some attributes of this R Script
patch_sql(id, **kwargs)	Update some attributes of this SQL script
post(credential_id, sql, name, ...)	Create a script
post_cancel(id)	Cancel a run
post_containers(docker_image_name, ...)	Create a container
post_containers_runs(id)	Start a run
post_custom(from_template_id, **kwargs)	Create a CustomScript
post_custom_runs(id)	Start a run
post_javascript(source, name, credential_id, ...)	Create a JavaScript Script
post_javascript_runs(id)	Start a run
post_python3(source, name, **kwargs)	Create a Python Script
post_python3_runs(id)	Start a run
post_r(source, name, **kwargs)	Create an R Script
post_r_runs(id)	Start a run
post_run(id)	Run a script
post_sql(name, credential_id, sql, ...)	Create a SQL script
post_sql_runs(id)	Start a run
put_containers(id, docker_image_name, ...)	Edit a container
put_containers_archive(id, status)	Update the archive status of this object
put_containers_projects(id, project_id)	Add a container docker to a project
put_containers_shares_groups(id, group_ids, ...)	Set the permissions groups has on this object
put_containers_shares_users(id, user_ids, ...)	Set the permissions users have on this object
put_custom(id, **kwargs)	Replace all attributes of this CustomScript
put_custom_archive(id, status)	Update the archive status of this object
put_custom_projects(id, project_id)	Add a Job to a project
put_custom_shares_groups(id, group_ids, ...)	Set the permissions groups has on this object
put_custom_shares_users(id, user_ids, ...)	Set the permissions users have on this object
put_javascript(id, source, name, ...)	Replace all attributes of this JavaScript Script
put_javascript_archive(id, status)	Update the archive status of this object
put_javascript_projects(id, project_id)	Add a scripted sql to a project
put_javascript_shares_groups(id, group_ids, ...)	Set the permissions groups has on this object
put_javascript_shares_users(id, user_ids, ...)	Set the permissions users have on this object

Continued on next page

Table 4.13 – continued from previous page

put_python3(id, source, name, **kwargs)	Replace all attributes of this Python Script
put_python3_archive(id, status)	Update the archive status of this object
put_python3_projects(id, project_id)	Add a python docker to a project
put_python3_shares_groups(id, group_ids, ...)	Set the permissions groups has on this object
put_python3_shares_users(id, user_ids, ...)	Set the permissions users have on this object
put_r(id, source, name, **kwargs)	Replace all attributes of this R Script
put_r_archive(id, status)	Update the archive status of this object
put_r_projects(id, project_id)	Add a r docker to a project
put_r_shares_groups(id, group_ids, ...)	Set the permissions groups has on this object
put_r_shares_users(id, user_ids, ...)	Set the permissions users have on this object
put_sql(id, name, credential_id, sql, ...)	Replace all attributes of this SQL script
put_sql_archive(id, status)	Update the archive status of this object
put_sql_projects(id, project_id)	Add a scripts to a project
put_sql_shares_groups(id, group_ids, ...)	Set the permissions groups has on this object
put_sql_shares_users(id, user_ids, ...)	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 **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

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.

updated_at : string/time

The time this script was last updated.

archived : string

The archival status of the requested object(s).

type : string

The type of script.

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
```

```
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

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

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

sql : string

The raw SQL query for the script.

id : integer

The ID for the script.

schedule : dict:

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

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.
```

get_containers(*id*)

View a container

Parameters **id** : integer

The ID for the script.

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

remote_host_credential_id : integer

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

docker_image_tag : string

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

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

repo_ref : string

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

type : string

The type of the script (e.g Container)

name : string

The name of the container.

git_credential_id : integer

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

running_as : dict:

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

from_template_id : integer

The ID of the template script.

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

docker_image_name : string

The name of the docker image to pull from DockerHub.

arguments : dict

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

notifications : dict:

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

author : dict:

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

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.

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]

template_script_name : string

The name of the template script.

repo_http_uri : string

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

id : integer

The ID for the script.

schedule : dict:

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

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.
```

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 **finished_at** : string/time

The time the last run completed.

state : string

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

id : integer

The ID of the run.

container_id : integer

The ID of the container.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

get_custom(*id*)

Get a CustomScript

Parameters **id** : integer

Returns **last_run** : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

credential_id : integer

The credential that this script will use.

type : string

The type of the script (e.g Custom)

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

The ID of the template script.

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

arguments : dict

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

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
```

```
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

author : dict:

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

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.

template_script_name : string

The name of the template script.

remote_host_id : integer

The remote host ID that this script will connect to.

id : integer

The ID for the script.

schedule : dict:

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

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

code_preview : string

The code that this script will run with arguments inserted.

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 **finished_at** : string/time

The time the last run completed.

state : string

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

custom_id : integer

The ID of the custom.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

get_javascript(*id*)

Get a JavaScript Script

Parameters **id** : integer

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

credential_id : integer

The credential that this script will use.

remote_host_id : integer

The remote host ID that this script will connect to.

id : integer

The ID for the script.

schedule : dict:

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

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.
```

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 **finished_at** : string/time

The time the last run completed.

state : string

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

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.

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 **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

```
- finished_at : string/time
    The time that the run completed.
- state : string
- id : integer
- error : string
```

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

```
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

id : integer

The ID for the script.

schedule : dict:

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

```
    Minutes of the day it is scheduled on
```

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.
```

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 **finished_at** : string/time

The time the last run completed.

state : string

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

python_id : integer

The ID of the python.

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.

started_at : string/time

The time the last run started at.

get_r(*id*)

Get an R Script

Parameters **id** : integer

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

```
- name : string
    The name of the project.
```

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

id : integer

The ID for the script.

schedule : dict:

```
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
```

```
    Hours of the day it is scheduled on
- scheduled : boolean
    If the object is scheduled
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per␣
↪hour
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

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.
```

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 **finished_at** : string/time

The time the last run completed.

state : string

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

r_id : integer

The ID of the r.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

get_sql(*id*)

Get a SQL script

Parameters **id** : integer

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

credential_id : integer

The credential that this script will use.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
```

```
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

sql : string

The raw SQL query for the script.

id : integer

The ID for the script.

code_preview : string

The code that this script will run with arguments inserted.

schedule : dict:

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

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per␣
↪hour
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

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.
```

remote_host_id : integer

The remote host ID that this script will connect to.

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 **finished_at** : string/time

The time that this run finished.

state : string

The state of this run.

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.
```

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.

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 **is_template** : boolean

Whether others scripts use this one as a template.

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

last_run : dict:

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

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

type : string

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

author : dict:

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

projects : list:

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

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

template_script_id : integer

The ID of the template script, if any.

archived : string

The archival status of the requested object(s).

parent_id : integer

The ID of the parent job that will trigger this script

name : string

The name of the script.

id : integer

The ID for the script.

from_template_id : integer

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

links : dict:

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

list_containers_projects(*id*)

List the projects a container docker belongs to

Parameters **id** : integer

The ID of the resource.

Returns **description** : string

A description of the project

created_at : string/time

archived : string

The archival status of the requested object(s).

name : string

The name of this project.

auto_share : boolean

id : integer

The ID for this project.

users : list:

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

updated_at : string/time

author : dict:

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

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 **finished_at** : string/time

The time the last run completed.

state : string

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

id : integer

The ID of the run.

container_id : integer

The ID of the container.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

`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 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.

message : string

The log message.

`list_containers_shares`(*id*)

List users and groups permissioned on this object

Parameters id : integer

The ID of the object.

Returns writers : 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.

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
```

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 **finished_at** : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

last_run : dict:

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

type : string

The type of the script (e.g Custom)

author : dict:

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

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.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

parent_id : integer

The ID of the parent job that will trigger this script

name : string

The name of the script.

id : integer

The ID for 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 description : string

A description of the project

created_at : string/time

archived : string

The archival status of the requested object(s).

name : string

The name of this project.

auto_share : boolean

id : integer

The ID for this project.

users : list:

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

updated_at : string/time

author : dict:

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

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 **finished_at** : string/time

The time the last run completed.

state : string

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

custom_id : integer

The ID of the custom.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

`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 **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.

message : string

The log message.

list_custom_shares(*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **writers** : 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.

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
```

list_history(*id*)

Get the run history and outputs of this script

Parameters **id** : integer

The ID for the script.

Returns **finished_at** : string/time

The time that this run finished.

state : string

The state of this run.

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.
```

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.

list_javascript_projects(*id*)

List the projects a scripted sql belongs to

Parameters **id** : integer

The ID of the resource.

Returns **description** : string

A description of the project

created_at : string/time

archived : string

The archival status of the requested object(s).

name : string

The name of this project.

auto_share : boolean

id : integer

The ID for this project.

users : list:

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

updated_at : string/time

author : dict:

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

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 **finished_at** : string/time

The time the last run completed.

state : string

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

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.

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 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.

message : string

The log message.

list_javascript_shares(*id*)

List users and groups permissioned on this object

Parameters id : integer

The ID of the object.

Returns writers : 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.

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
```

list_python3_projects(*id*)

List the projects a python docker belongs to

Parameters id : integer

The ID of the resource.

Returns description : string

A description of the project

created_at : string/time

archived : string

The archival status of the requested object(s).

name : string

The name of this project.

auto_share : boolean

id : integer

The ID for this project.

users : list:

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

updated_at : string/time

author : dict:

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

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 **finished_at** : string/time

The time the last run completed.

state : string

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

python_id : integer

The ID of the python.

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.

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 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.

message : string

The log message.

list_python3_shares(*id*)

List users and groups permissioned on this object

Parameters id : integer

The ID of the object.

Returns writers : 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.

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
```

list_r_projects(*id*)

List the projects a r docker belongs to

Parameters **id** : integer

The ID of the resource.

Returns **description** : string

A description of the project

created_at : string/time

archived : string

The archival status of the requested object(s).

name : string

The name of this project.

auto_share : boolean

id : integer

The ID for this project.

users : list:

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

updated_at : string/time

author : dict:

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

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 **finished_at** : string/time

The time the last run completed.

state : string

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

r_id : integer

The ID of the r.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

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 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.

message : string

The log message.

list_r_shares(*id*)

List users and groups permissioned on this object

Parameters id : integer

The ID of the object.

Returns writers : 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.

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
```

list_sql_projects(*id*)

List the projects a scripts belongs to

Parameters **id** : integer

The ID of the resource.

Returns **description** : string

A description of the project

created_at : string/time

archived : string

The archival status of the requested object(s).

name : string

The name of this project.

auto_share : boolean

id : integer

The ID for this project.

users : list:

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

updated_at : string/time

author : dict:

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

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 **finished_at** : string/time

The time that this run finished.

state : string

The state of this run.

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.
```

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.

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 **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.

message : string

The log message.

list_sql_shares(*id*)

List users and groups permissioned on this object

Parameters **id** : integer

The ID of the object.

Returns **writers** : 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.

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
```

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.

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.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

sql : string, optional

The raw SQL query for the script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

arguments : dict, optional

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

notifications : dict, optional:

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

template_script_id : integer, optional

The ID of the template script, if any. A script cannot both have a template script and be a template for other scripts.

name : string, optional

The name of the script.

schedule : dict, optional:

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

```
    Minutes of the day it is scheduled on
```

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

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.

updated_at : string/time

The time this script was last updated.

archived : string

The archival status of the requested object(s).

type : string

The type of script.

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

sql : string

The raw SQL query for the script.

id : integer

The ID for the script.

schedule : dict:

```
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
```

```
    Hours of the day it is scheduled on
- scheduled : boolean
    If the object is scheduled
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per␣
↪hour
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

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.
```

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

Update a container

Parameters id : integer

The ID for the script.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

docker_image_name : string, optional

The name of the docker image to pull from DockerHub.

remote_host_credential_id : integer, optional

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

repo_ref : string, optional

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

arguments : dict, optional

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

notifications : dict, optional:

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

required_resources : dict, optional:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

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]

target_project_id : integer, optional

Target project to which script outputs will be added.

repo_http_uri : string, optional

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

name : string, optional

The name of the container.

git_credential_id : integer, optional

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

docker_image_tag : string, optional

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

schedule : dict, optional:

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

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

remote_host_credential_id : integer

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

docker_image_tag : string

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

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

repo_ref : string

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

type : string

The type of the script (e.g Container)

name : string

The name of the container.

git_credential_id : integer

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

running_as : dict:

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

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

from_template_id : integer

The ID of the template script.

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

docker_image_name : string

The name of the docker image to pull from DockerHub.

arguments : dict

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

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
```

```
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

author : dict:

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

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.

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]

template_script_name : string

The name of the template script.

repo_http_uri : string

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

id : integer

The ID for the script.

schedule : dict:

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

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.
```

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

Update some attributes of this CustomScript

Parameters **id** : integer

The ID for the script.

credential_id : integer, optional

The credential that this script will use.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

parent_id : integer, optional

The ID of the parent job that will trigger this script

arguments : dict, optional

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

target_project_id : integer, optional

Target project to which script outputs will be added.

notifications : dict, optional:

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

name : string, optional

The name of the script.

schedule : dict, optional:

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

Returns **last_run** : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

credential_id : integer

The credential that this script will use.

type : string

The type of the script (e.g Custom)

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

The ID of the template script.

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

arguments : dict

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

notifications : dict:

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

author : dict:

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

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.

template_script_name : string

The name of the template script.

remote_host_id : integer

The remote host ID that this script will connect to.

id : integer

The ID for the script.

schedule : dict:

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

```
- scheduled : boolean
    If the object is scheduled
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per␣
↪hour
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

code_preview : string

The code that this script will run with arguments inserted.

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

Update some attributes of this JavaScript Script

Parameters **id** : integer

The ID for the script.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

source : string, optional

The body/text of the script.

arguments : dict, optional

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

notifications : dict, optional:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
```

```
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

name : string, optional

The name of the script.

time_zone : string, optional

The time zone of this script.

user_context : string, optional

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

credential_id : integer, optional

The credential that this script will use.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

parent_id : integer, optional

The ID of the parent job that will trigger this script

next_run_at : string/time, optional

The time of the next scheduled run.

target_project_id : integer, optional

Target project to which script outputs will be added.

schedule : dict, optional:

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

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
```

```
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

credential_id : integer

The credential that this script will use.

remote_host_id : integer

The remote host ID that this script will connect to.

id : integer

The ID for the script.

schedule : dict:

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

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per␣
↪hour
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

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.
```

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

Update some attributes of this Python Script

Parameters **id** : integer

The ID for the script.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

source : string, optional

The body/text of the script.

arguments : dict, optional

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

notifications : dict, optional:

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

```
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

required_resources : dict, optional:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

name : string, optional

The name of the script.

time_zone : string, optional

The time zone of this script.

user_context : string, optional

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

next_run_at : string/time, optional

The time of the next scheduled run.

target_project_id : integer, optional

Target project to which script outputs will be added.

schedule : dict, optional:

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

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
```

```
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

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

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

id : integer

The ID for the script.

schedule : dict:

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

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.
```

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

Update some attributes of this R Script

Parameters **id** : integer

The ID for the script.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

source : string, optional

The body/text of the script.

arguments : dict, optional

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

notifications : dict, optional:

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

required_resources : dict, optional:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
```

```
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

name : string, optional

The name of the script.

time_zone : string, optional

The time zone of this script.

user_context : string, optional

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

next_run_at : string/time, optional

The time of the next scheduled run.

target_project_id : integer, optional

Target project to which script outputs will be added.

schedule : dict, optional:

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

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

id : integer

The ID for the script.

schedule : dict:

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

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per␣
↪hour
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

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.
```

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

Update some attributes of this SQL script

Parameters id : integer

The ID for the script.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

arguments : dict, optional

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

notifications : dict, optional:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
```

```
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

name : string, optional

The name of the script.

time_zone : string, optional

The time zone of this script.

user_context : string, optional

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

credential_id : integer, optional

The credential that this script will use.

sql : string, optional

The raw SQL query for the script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

next_run_at : string/time, optional

The time of the next scheduled run.

target_project_id : integer, optional

Target project to which script outputs will be added.

schedule : dict, optional:

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

remote_host_id : integer, optional

The remote host ID that this script will connect to.

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

credential_id : integer

The credential that this script will use.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

sql : string

The raw SQL query for the script.

id : integer

The ID for the script.

code_preview : string

The code that this script will run with arguments inserted.

schedule : dict:

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

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.
```

remote_host_id : integer

The remote host ID that this script will connect to.

post(*credential_id*, *sql*, *name*, *remote_host_id*, ***kwargs*)

Create a script

Parameters **credential_id** : integer

The credential ID.

sql : string

The raw SQL query for the script.

name : string

The name of the script.

remote_host_id : integer

The database ID.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
Cannot be set if this script uses a template script.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

notifications : dict, optional:

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

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.

hidden : boolean, optional

The hidden status of the object.

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

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.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

id : integer

The ID for the script.

schedule : dict:

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

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.
```

post_cancel(*id*)

Cancel a run

Parameters **id** : integer

The ID of the job.

Returns **id** : integer

The ID of the run.

state : string

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

is_cancel_requested : boolean

True if run cancel requested, else false.

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

Create a container

Parameters **docker_image_name** : string

The name of the docker image to pull from DockerHub.

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

docker_command : string

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

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
```

```
- required : boolean
    Whether this param is required.
```

remote_host_credential_id : integer, optional

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

repo_ref : string, optional

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

arguments : dict, optional

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

notifications : dict, optional:

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

hidden : boolean, optional

The hidden status of the object.

user_context : string, optional

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

target_project_id : integer, optional

Target project to which script outputs will be added.

repo_http_uri : string, optional

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

name : string, optional

The name of the container.

git_credential_id : integer, optional

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

docker_image_tag : string, optional

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

schedule : dict, optional:

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

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

remote_host_credential_id : integer

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
```

```
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

docker_image_tag : string

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

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

repo_ref : string

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

type : string

The type of the script (e.g Container)

name : string

The name of the container.

git_credential_id : integer

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

running_as : dict:

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

from_template_id : integer

The ID of the template script.

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

docker_image_name : string

The name of the docker image to pull from DockerHub.

arguments : dict

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

notifications : dict:

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

author : dict:

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

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.

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]

template_script_name : string

The name of the template script.

repo_http_uri : string

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

id : integer

The ID for the script.

schedule : dict:

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

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

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.
```

post_containers_runs(*id*)

Start a run

Parameters **id** : integer

The ID of the container.

Returns **finished_at** : string/time

The time the last run completed.

state : string

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

id : integer

The ID of the run.

container_id : integer

The ID of the container.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

post_custom(*from_template_id*, ***kwargs*)

Create a CustomScript

Parameters **from_template_id** : integer

The ID of the template script.

credential_id : integer, optional

The credential that this script will use.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

parent_id : integer, optional

The ID of the parent job that will trigger this script

arguments : dict, optional

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

target_project_id : integer, optional

Target project to which script outputs will be added.

notifications : dict, optional:

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

name : string, optional

The name of the script.

hidden : boolean, optional

The hidden status of the object.

schedule : dict, optional:

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

Returns **last_run** : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

credential_id : integer

The credential that this script will use.

type : string

The type of the script (e.g Custom)

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

The ID of the template script.

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

arguments : dict

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

notifications : dict:

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

author : dict:

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

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.

template_script_name : string

The name of the template script.

remote_host_id : integer

The remote host ID that this script will connect to.

id : integer

The ID for the script.

schedule : dict:

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

code_preview : string

The code that this script will run with arguments inserted.

post_custom_runs(*id*)

Start a run

Parameters **id** : integer

The ID of the custom.

Returns **finished_at** : string/time

The time the last run completed.

state : string

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

custom_id : integer

The ID of the custom.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

post_javascript(*source*, *name*, *credential_id*, *remote_host_id*, ***kwargs*)

Create a JavaScript Script

Parameters **source** : string

The body/text of the script.

name : string

The name of the script.

credential_id : integer

The credential that this script will use.

remote_host_id : integer

The remote host ID that this script will connect to.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

arguments : dict, optional

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

notifications : dict, optional:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes␣
↪successfully.
```

```
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

hidden : boolean, optional

The hidden status of the object.

user_context : string, optional

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

time_zone : string, optional

The time zone of this script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

next_run_at : string/time, optional

The time of the next scheduled run.

target_project_id : integer, optional

Target project to which script outputs will be added.

schedule : dict, optional:

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

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

credential_id : integer

The credential that this script will use.

remote_host_id : integer

The remote host ID that this script will connect to.

id : integer

The ID for the script.

schedule : dict:

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

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.
```

post_javascript_runs(*id*)

Start a run

Parameters **id** : integer

The ID of the javascript.

Returns **finished_at** : string/time

The time the last run completed.

state : string

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

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.

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.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
```

```
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

arguments : dict, optional

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

notifications : dict, optional:

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

required_resources : dict, optional:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

hidden : boolean, optional

The hidden status of the object.

user_context : string, optional

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

time_zone : string, optional

The time zone of this script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

next_run_at : string/time, optional

The time of the next scheduled run.

target_project_id : integer, optional

Target project to which script outputs will be added.

schedule : dict, optional:

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

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

```
- 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
```

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

id : integer

The ID for the script.

schedule : dict:

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

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.
```

post_python3_runs (*id*)
Start a run

Parameters **id** : integer

The ID of the python.

Returns **finished_at** : string/time

The time the last run completed.

state : string

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

python_id : integer

The ID of the python.

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.

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.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

arguments : dict, optional

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

notifications : dict, optional:

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

required_resources : dict, optional:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

hidden : boolean, optional

The hidden status of the object.

user_context : string, optional

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

time_zone : string, optional

The time zone of this script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

next_run_at : string/time, optional

The time of the next scheduled run.

target_project_id : integer, optional

Target project to which script outputs will be added.

schedule : dict, optional:

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

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

id : integer

The ID for the script.

schedule : dict:

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

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.
```

post_r_runs(*id*)

Start a run

Parameters **id** : integer

The ID of the r.

Returns **finished_at** : string/time

The time the last run completed.

state : string

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

r_id : integer

The ID of the r.

id : integer

The ID of the run.

error : string

The error, if any, returned by the run.

started_at : string/time

The time the last run started at.

is_cancel_requested : boolean

True if run cancel requested, else false.

post_run(*id*)

Run a script

Parameters **id** : integer

The ID for the script.

Returns None

Response code 204: success

post_sql(*name*, *credential_id*, *sql*, *remote_host_id*, ***kwargs*)

Create a SQL script

Parameters **name** : string

The name of the script.

credential_id : integer

The credential that this script will use.

sql : string

The raw SQL query for the script.

remote_host_id : integer

The remote host ID that this script will connect to.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

arguments : dict, optional

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

notifications : dict, optional:

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

hidden : boolean, optional

The hidden status of the object.

user_context : string, optional

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

time_zone : string, optional

The time zone of this script.

parent_id : integer, optional

The ID of the parent job that will trigger this script

next_run_at : string/time, optional

The time of the next scheduled run.

target_project_id : integer, optional

Target project to which script outputs will be added.

schedule : dict, optional:

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

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

credential_id : integer

The credential that this script will use.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
```

```
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

sql : string

The raw SQL query for the script.

id : integer

The ID for the script.

code_preview : string

The code that this script will run with arguments inserted.

schedule : dict:

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

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.
```

remote_host_id : integer

The remote host ID that this script will connect to.

post_sql_runs(*id*)

Start a run

Parameters **id** : integer

The ID of the sql.

Returns **finished_at** : string/time

The time that this run finished.

state : string

The state of this run.

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.
```

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.

started_at : string/time

The time the last run started.

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

Edit a container

Parameters **id** : integer

The ID for the script.

docker_image_name : string

The name of the docker image to pull from DockerHub.

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

docker_command : string

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

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

remote_host_credential_id : integer, optional

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

repo_ref : string, optional

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

arguments : dict, optional

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

notifications : dict, optional:

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

user_context : string, optional

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

target_project_id : integer, optional

Target project to which script outputs will be added.

repo_http_uri : string, optional

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

name : string, optional

The name of the container.

git_credential_id : integer, optional

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

docker_image_tag : string, optional

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

schedule : dict, optional:

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

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

remote_host_credential_id : integer

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

required_resources : dict:

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

```
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

docker_image_tag : string

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

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

repo_ref : string

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

type : string

The type of the script (e.g Container)

name : string

The name of the container.

git_credential_id : integer

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

running_as : dict:

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

from_template_id : integer

The ID of the template script.

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

docker_image_name : string

The name of the docker image to pull from DockerHub.

arguments : dict

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

notifications : dict:

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

author : dict:

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

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.

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]

template_script_name : string

The name of the template script.

repo_http_uri : string

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

id : integer

The ID for the script.

schedule : dict:

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

```
    Alternative to scheduled minutes, number of times to run per␣
↪hour
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

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.
```

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 **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

remote_host_credential_id : integer

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB).
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
```

```
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

docker_image_tag : string

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

target_project_id : integer

Target project to which script outputs will be added.

archived : string

The archival status of the requested object(s).

repo_ref : string

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

type : string

The type of the script (e.g Container)

name : string

The name of the container.

git_credential_id : integer

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

running_as : dict:

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

from_template_id : integer

The ID of the template script.

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

docker_image_name : string

The name of the docker image to pull from DockerHub.

arguments : dict

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

notifications : dict:

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

author : dict:

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

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.

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]

template_script_name : string

The name of the template script.

repo_http_uri : string

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

id : integer

The ID for the script.

schedule : dict:

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

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

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.
```

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 **writers** : 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.

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
```

put_containers_shares_users(*id*, *user_ids*, *permission_level*)

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

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

Returns **writers** : 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.

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
```

put_custom(*id*, ***kwargs*)

Replace all attributes of this CustomScript

Parameters **id** : integer

The ID for the script.

credential_id : integer, optional

The credential that this script will use.

remote_host_id : integer, optional

The remote host ID that this script will connect to.

parent_id : integer, optional

The ID of the parent job that will trigger this script

arguments : dict, optional

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

target_project_id : integer, optional

Target project to which script outputs will be added.

notifications : dict, optional:

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

name : string, optional

The name of the script.

schedule : dict, optional:

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

```
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per␣
↪hour
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

Returns **last_run** : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

credential_id : integer

The credential that this script will use.

type : string

The type of the script (e.g Custom)

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

The ID of the template script.

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

arguments : dict

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

notifications : dict:

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

author : dict:

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

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.

template_script_name : string

The name of the template script.

remote_host_id : integer

The remote host ID that this script will connect to.

id : integer

The ID for the script.

schedule : dict:

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

code_preview : string

The code that this script will run with arguments inserted.

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 **last_run** : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

credential_id : integer

The credential that this script will use.

type : string

The type of the script (e.g Custom)

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

The ID of the template script.

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

arguments : dict

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

notifications : dict:

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

author : dict:

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

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.

template_script_name : string

The name of the template script.

remote_host_id : integer

The remote host ID that this script will connect to.

id : integer

The ID for the script.

schedule : dict:

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

code_preview : string

The code that this script will run with arguments inserted.

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 **writers** : 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.

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
```

put_custom_shares_users(*id*, *user_ids*, *permission_level*)

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

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

Returns **writers** : 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.

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
```

put_javascript(*id*, *source*, *name*, *credential_id*, *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.

name : string

The name of the script.

credential_id : integer

The credential that this script will use.

remote_host_id : integer

The remote host ID that this script will connect to.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

arguments : dict, optional

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

notifications : dict, optional:

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

time_zone : string, optional

The time zone of this script.

user_context : string, optional

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

next_run_at : string/time, optional

The time of the next scheduled run.

target_project_id : integer, optional

Target project to which script outputs will be added.

schedule : dict, optional:

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

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

credential_id : integer

The credential that this script will use.

remote_host_id : integer

The remote host ID that this script will connect to.

id : integer

The ID for the script.

schedule : dict:

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

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.
```

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 **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
```

```
- urls : list
    URLs to receive a POST request at job completion
- 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
```

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

credential_id : integer

The credential that this script will use.

remote_host_id : integer

The remote host ID that this script will connect to.

id : integer

The ID for the script.

schedule : dict:

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

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

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.
```

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 **writers** : 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.

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
```

put_javascript_shares_users(*id*, *user_ids*, *permission_level*)

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

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

Returns **writers** : 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.

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
```

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.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

arguments : dict, optional

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

notifications : dict, optional:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
```

```
- urls : list
    URLs to receive a POST request at job completion
- 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
```

required_resources : dict, optional:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

time_zone : string, optional

The time zone of this script.

user_context : string, optional

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

next_run_at : string/time, optional

The time of the next scheduled run.

target_project_id : integer, optional

Target project to which script outputs will be added.

schedule : dict, optional:

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

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

id : integer

The ID for the script.

schedule : dict:

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

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.
```

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 **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

```
- finished_at : string/time
    The time that the run completed.
- state : string
- id : integer
- error : string
```

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

```
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

id : integer

The ID for the script.

schedule : dict:

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

```
    Minutes of the day it is scheduled on
```

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.
```

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 **writers** : 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.

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
```

put_python3_shares_users(*id*, *user_ids*, *permission_level*)

Set the permissions users have on this object

Parameters id : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

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

Returns writers : 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.

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
```

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.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

arguments : dict, optional

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

notifications : dict, optional:

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

```
- 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
```

required_resources : dict, optional:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

time_zone : string, optional

The time zone of this script.

user_context : string, optional

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

next_run_at : string/time, optional

The time of the next scheduled run.

target_project_id : integer, optional

Target project to which script outputs will be added.

schedule : dict, optional:

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

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
```

```
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

id : integer

The ID for the script.

schedule : dict:

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

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.
```

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 **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

required_resources : dict:

```
- cpu : integer
    The number of CPU shares to allocate for the container. Each␣
↪core has
    1024 shares. Must be at least 2 shares.
- memory : integer
    The amount of RAM to allocate for the container (in MiB). Must␣
↪be at
    least 4 MiB.
- disk_space : number/float
    The amount of disk space, in GB, to allocate for the container.␣
↪This
    space will be used to hold the git repo configured for the␣
↪container
    and anything your container writes to /tmp or /data. Fractional␣
↪values
    (e.g. 0.25) are supported.
```

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

source : string

The body/text of the script.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

running_as : dict:

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
```

```
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

author : dict:

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

arguments : dict

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

hidden : boolean

The hidden status of the object.

user_context : string

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

template_script_name : string

The name of the template script.

id : integer

The ID for the script.

schedule : dict:

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

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.
```

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 **writers** : 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.

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
```

put_r_shares_users(*id*, *user_ids*, *permission_level*)

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

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

Returns **writers** : 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.

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
```

put_sql (*id*, *name*, *credential_id*, *sql*, *remote_host_id*, ***kwargs*)

Replace all attributes of this SQL script

Parameters **id** : integer

The ID for the script.

name : string

The name of the script.

credential_id : integer

The credential that this script will use.

sql : string

The raw SQL query for the script.

remote_host_id : integer

The remote host ID that this script will connect to.

params : list, optional:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

arguments : dict, optional

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

notifications : dict, optional:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes␣
↪successfully.
```

```
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

time_zone : string, optional

The time zone of this script.

user_context : string, optional

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

parent_id : integer, optional

The ID of the parent job that will trigger this script

next_run_at : string/time, optional

The time of the next scheduled run.

target_project_id : integer, optional

Target project to which script outputs will be added.

schedule : dict, optional:

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

Returns **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

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

last_run : dict:

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

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

credential_id : integer

The credential that this script will use.

type : string

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

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

running_as : dict:

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

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

from_template_id : integer

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

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

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

author : dict:

```
- id : integer
    The ID of this user.
- username : string
    This user's username.
- online : boolean
    Whether this user is online.
- initials : string
    This user's initials.
- name : string
    This user's name.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

hidden : boolean

The hidden status of the object.

user_context : string

"runner" or "author", who to execute the script as when run as a template.

template_script_name : string

The name of the template script.

sql : string

The raw SQL query for the script.

id : integer

The ID for the script.

code_preview : string

The code that this script will run with arguments inserted.

schedule : dict:

```
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled : boolean
    If the object is scheduled
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per␣
↪hour
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

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.
```

remote_host_id : integer

The remote host ID that this script will connect to.

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 **is_template** : boolean

Whether others scripts use this one as a template.

published_as_template_id : integer

The ID of the template that this script is backing.

projects : list:

```
A list of projects containing the script.
- id : integer
    The ID for the project.
- name : string
    The name of the project.
```

last_run : dict:

```
- finished_at : string/time
    The time that the run completed.
- state : string
- id : integer
- error : string
    The error message for this run, if present.
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
```

parent_id : integer

The ID of the parent job that will trigger this script

next_run_at : string/time

The time of the next scheduled run.

time_zone : string

The time zone of this script.

created_at : string/time

The time this script was created.

updated_at : string/time

The time the script was last updated.

archived : string

The archival status of the requested object(s).

credential_id : integer

The credential that this script will use.

type : string

The type of the script (e.g SQL, Container, Python, R, JavaScript)

name : string

The name of the script.

target_project_id : integer

Target project to which script outputs will be added.

expanded_arguments : dict

Expanded arguments for use in injecting into different environments.

running_as : dict:

```
- id : integer
    The ID of this user.
- username : string
    This user's username.
- online : boolean
    Whether this user is online.
- initials : string
    This user's initials.
- name : string
    This user's name.
```

from_template_id : integer

The ID of the template this script uses, if any.

params : list:

```
A definition of the parameters this script accepts in the arguments␣
↪field.
- description : string
    A short sentence or fragment describing this parameter to the␣
↪end user.
- 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.
- label : string
    The label to present to users when asking them for the value.
- 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.
- required : boolean
    Whether this param is required.
```

finished_at : string/time

The time that the script's last run finished.

state : string

The status of the script's last run.

template_dependents_count : integer

How many other scripts use this one as a template.

notifications : dict:

```
- success_email_body : string
    Custom body text for success e-mail, written in Markdown.
- stall_warning_minutes : integer
    Stall warning emails will be sent after this amount of minutes.
- success_email_addresses : list
    Addresses to notify by e-mail when the job completes␣
↪successfully.
- success_email_subject : string
    Custom subject line for success e-mail.
- urls : list
    URLs to receive a POST request at job completion
- 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
```

author : dict:

```
- id : integer
    The ID of this user.
- username : string
    This user's username.
- online : boolean
    Whether this user is online.
- initials : string
    This user's initials.
- name : string
    This user's name.
```

arguments : dict

Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params.

hidden : boolean

The hidden status of the object.

user_context : string

"runner" or "author", who to execute the script as when run as a template.

template_script_name : string

The name of the template script.

sql : string

The raw SQL query for the script.

id : integer

The ID for the script.

code_preview : string

The code that this script will run with arguments inserted.

schedule : dict:

```
- scheduled_days : list
    Day based on numeric value starting at 0 for Sunday
- scheduled_hours : list
    Hours of the day it is scheduled on
- scheduled : boolean
    If the object is scheduled
- scheduled_runs_per_hour : integer
    Alternative to scheduled minutes, number of times to run per␣
↪hour
- scheduled_minutes : list
    Minutes of the day it is scheduled on
```

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.
```

remote_host_id : integer

The remote host ID that this script will connect to.

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 **writers** : 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.

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
```

put_sql_shares_users(*id*, *user_ids*, *permission_level*)

Set the permissions users have on this object

Parameters **id** : integer

ID of the resource to be shared

user_ids : list

An array of one or more user IDs

permission_level : string

Options are: "read", "write", or "manage"

Returns **writers** : 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.

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
```

Tables

class `Tables` (*session*, *return_type='civis'*)

Methods

get(id)	Show basic table info
get_enhancements_cass_ncoa(id, source_table_id)	View the status of a CASS / NCOA table enhancement
get_enhancements_geocodings(id, source_table_id)	View the status of a geocoding table enhancement
get_enhancements_prepared_matchings(id, ...)	View a prepared matching enhancement
get_enhancements_table_matchings(id, ...)	View a table matching enhancement
list(**kwargs)	List tables
list_columns(id, **kwargs)	List columns in the specified table
patch(id, **kwargs)	Update a table
post(data, name, schema, database_id)	Import a file into a table
post_enhancements_cass_ncoa(source_table_id, ...)	Standardize addresses in a table
post_enhancements_geocodings(source_table_id)	Geocode a table

Continued on next page

Table 4.14 – continued from previous page

post_enhancements_prepared_matchings(...)	Match person records against a dynamo table prepared by Civis
post_enhancements_table_matchings(...)	Match person records against an arbitrary Redshift table
post_refresh(id)	Request a refresh for column and table statistics

get(*id*)

Show basic table info

Parameters **id** : integer

Returns **owner** : string

The database username of the table's owner.

last_run : dict:

```
- finished_at : string/time
    The time that the run completed.
- state : string
- id : integer
- error : string
    The error message for this run, if present.
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
```

column_count : integer

The number of columns in the table.

database_id : integer

The ID of the database.

refresh_id : string

The ID of the most recent statistics refresh.

schema : string

The name of the schema containing the table.

name : string

Name of the table.

view_def : string

columns : list:

```
- useable_as_primary_key : boolean
    Whether the column may be used as an primary key to identify␣
↪table
    rows.
- 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.
- sql_type : string
    SQL type of the column.
- min_value : string
```

```
    Smallest value in the column.
- useable_as_independent_variable : boolean
    Whether the column may be used as an independent variable to␣
↪train a
    model.
- value_distribution : dict
    An object mapping distinct values in the column to the number␣
↪of times
    they appear in the column
- distinct_count : integer
    Number of distinct values in the column.
- order : integer
    Relative position of the column in the table.
- avg_value : number/float
    Average value of the column, where applicable.
- description : string
    The description of the column, as specified by the table owner
- null_count : integer
    Number of null values in the column.
- encoding : string
    The compression encoding for this columnSee: http://docs.aws.
↪amazon.com
    /redshift/latest/dg/c_Compression_encodings.html
- sample_values : list
    A sample of values from the column.
- name : string
    Name of the column.
- max_value : string
    Largest value in the column.
- stddev : number/float
    Stddev of the column, where applicable.
- coverage_count : integer
    Number of non-null 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.
```

ontology_mapping : dict

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

id : integer

The ID of the table.

row_count : integer

The number of rows in the table.

multipart_key : list

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.

outgoing_table_matches : list:

```
- source_table_id : integer
    Source table
- target : dict::
    - name : string
- target_id : integer
    Target ID
- target_type : string
    Target type
- job : dict::
    - match_options : dict::
        - max_matches : integer
        - threshold : string
    - state : string
        Whether the job is idle, queued, running, cancelled, or␣
↪failed.
    - runs : list::
        Information about the most recent runs of the job.
        - finished_at : string/time
            The time that the run completed.
        - state : string
        - id : integer
        - error : string
            The error message for this run, if present.
        - created_at : string/time
            The time that the run was queued.
        - started_at : string/time
            The time that the run started.
    - type : string
    - name : string
    - last_run : dict::
        - finished_at : string/time
            The time that the run completed.
        - state : string
        - id : integer
        - error : string
            The error message for this run, if present.
        - created_at : string/time
            The time that the run was queued.
        - started_at : string/time
            The time that the run started.
    - id : integer
    - hidden : boolean
        The hidden status of the object.
    - created_at : string/date-time
    - updated_at : string/date-time
```

size_mb : number/float

The size of the table in megabytes.

description : string

The description of the table, as specified by the table owner

is_view : boolean

True if this table represents a view. False if it represents a regular table.

enhancements : list:

```
- updated_at : string/time
- join_id : integer
- created_at : string/time
- type : string
```

last_refresh : string/date-time

The time of the last statistics refresh.

sortkeys : string

The column used as the Amazon Redshift sortkey.

joins : list:

```
- left_table_id : integer
- right_table_id : integer
- left_identifier : string
- left_join : boolean
- id : integer
- right_identifier : string
- on : string
- updated_at : string/time
- created_at : string/time
```

distkey : string

The column used as the Amazon Redshift distkey.

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.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

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.

id : integer

The ID of the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

ncoa_credential_id : integer

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

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.

enhanced_table_name : string

The name of the table created by the enhancement.

id : integer

The ID of the enhancement.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

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.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

match_table_id : integer

The ID of the Dynamo table to match against.

id : integer

The ID of the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match. Must be less than or equal to 1 and greater than or equal to 0.

max_matches : integer

The maximum number of individuals a person may be matched with.A value of 0 indicates that all matches should be returned.

`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.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

match_table_id : integer

The ID of the Redshift table to match against.

id : integer

The ID of the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match. Must be less than or equal to 1 and greater than or equal to 0.

max_matches : integer

The maximum number of individuals a person may be matched with.A value of 0 indicates that all matches should be returned.

`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 **owner** : string

The database username of the table's owner.

row_count : integer

The number of rows in the table.

last_run : dict:

```
- finished_at : string/time
    The time that the run completed.
- state : string
- id : integer
- error : string
    The error message for this run, if present.
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
```

refresh_id : string

The ID of the most recent statistics refresh.

sortkeys : string

The column used as the Amazon Redshift sortkey.

refresh_status : string

How up-to-date the table's statistics on row counts, null counts, distinct counts, and values distributions are. One of: refreshing, stale, or current.

column_count : integer

The number of columns in the table.

database_id : integer

The ID of the database.

description : string

The description of the table, as specified by the table owner

is_view : boolean

True if this table represents a view. False if it represents a regular table.

schema : string

The name of the schema containing the table.

last_refresh : string/date-time

The time of the last statistics refresh.

name : string

Name of the table.

id : integer

The ID of the table.

size_mb : number/float

The size of the table in megabytes.

distkey : string

The column used as the Amazon Redshift distkey.

`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 **useable_as_primary_key** : boolean

Whether the column may be used as an primary key to identify table rows.

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.

sql_type : string

SQL type of the column.

min_value : string

Smallest value in the column.

useable_as_independent_variable : boolean

Whether the column may be used as an independent variable to train a model.

value_distribution : dict

An object mapping distinct values in the column to the number of times they appear in the column

distinct_count : integer

Number of distinct values in the column.

order : integer

Relative position of the column in the table.

avg_value : number/float

Average value of the column, where applicable.

description : string

The description of the column, as specified by the table owner

null_count : integer

Number of null values in the column.

encoding : string

The compression encoding for this columnSee: http://docs.aws.amazon.com/redshift/latest/dg/c_Compression_encodings.html

sample_values : list

A sample of values from the column.

name : string

Name of the column.

max_value : string

Largest value in the column.

stddev : number/float

Stddev of the column, where applicable.

coverage_count : integer

Number of non-null 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.

patch(*id*, ***kwargs*)

Update a table

Parameters id : integer

The ID of the table.

description : string, optional

The user-defined description of the table.

ontology_mapping : dict, optional

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

Returns owner : string

The database username of the table's owner.

row_count : integer

The number of rows in the table.

last_run : dict:

```
- finished_at : string/time
    The time that the run completed.
- state : string
- id : integer
- error : string
    The error message for this run, if present.
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
```

refresh_id : string

The ID of the most recent statistics refresh.

sortkeys : string

The column used as the Amazon Redshift sortkey.

refresh_status : string

How up-to-date the table's statistics on row counts, null counts, distinct counts, and values distributions are. One of: refreshing, stale, or current.

column_count : integer

The number of columns in the table.

database_id : integer

The ID of the database.

description : string

The description of the table, as specified by the table owner

is_view : boolean

True if this table represents a view. False if it represents a regular table.

schema : string

The name of the schema containing the table.

last_refresh : string/date-time

The time of the last statistics refresh.

name : string

Name of the table.

id : integer

The ID of the table.

ontology_mapping : dict

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

size_mb : number/float

The size of the table in megabytes.

distkey : string

The column used as the Amazon Redshift distkey.

post(*data*, *name*, *schema*, *database_id*)

Import a file into a table

Parameters **data** : string

The file to import, uploaded using HTTP multipart.

name : string

The destination table name, without the schema prefix.

schema : string

The destination schema name.

database_id : integer

The ID of the destination database.

Returns **finished_at** : string/date-time

The end time of the last run.

state : string

The state of the last run.

schema : string

The destination schema name.

name : string

The destination table name, without the schema prefix.

started_at : string/date-time

The start time of the last run.

database_id : integer

The ID of the destination database.

post_enhancements_cass_ncoa(*source_table_id*, ***kwargs*)

Standardize addresses in a table

Parameters **source_table_id** : integer

The ID of the table to be enhanced.

output_level : string, optional

The set of fields persisted by a CASS or NCOA enhancement.For CASS enhancements, one of 'cass' or 'all.'For NCOA enhancements, one of 'cass', 'ncoa' , 'coalesced' or 'all'.By default, all fields will be returned.

ncoa_credential_id : integer, optional

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

perform_ncoa : boolean, optional

Whether to update addresses for records matching the National Change of Address (NCOA) database.

Returns **source_table_id** : integer

The ID of the table that was enhanced.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

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.

id : integer

The ID of the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

ncoa_credential_id : integer

Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

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.

enhanced_table_name : string

The name of the table created by the enhancement.

id : integer

The ID of the enhancement.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

post_enhancements_prepared_matchings(*source_table_id*, *match_table_id*, *threshold*, ***kwargs*)

Match person records against a dynamo table prepared by Civis

Parameters **source_table_id** : integer

The ID of the table to be enhanced.

match_table_id : integer

The ID of the Dynamo table to match against.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match. Must be less than or equal to 1 and greater than or equal to 0.

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.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

match_table_id : integer

The ID of the Dynamo table to match against.

id : integer

The ID of the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match. Must be less than or equal to 1 and greater than or equal to 0.

max_matches : integer

The maximum number of individuals a person may be matched with.A value of 0 indicates that all matches should be returned.

post_enhancements_table_matchings(*source_table_id*, *match_table_id*, *threshold*, ***kwargs*)

Match person records against an arbitrary Redshift table

Parameters **source_table_id** : integer

The ID of the table to be enhanced.

match_table_id : integer

The ID of the Redshift table to match against.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match. Must be less than or equal to 1 and greater than or equal to 0.

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.

state : string

The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

enhanced_table_schema : string

The schema name of the table created by the enhancement.

match_table_id : integer

The ID of the Redshift table to match against.

id : integer

The ID of the enhancement.

enhanced_table_name : string

The name of the table created by the enhancement.

threshold : number/float

The confidence threshold which must be met for two individuals to be declared a match. Must be less than or equal to 1 and greater than or equal to 0.

max_matches : integer

The maximum number of individuals a person may be matched with.A value of 0 indicates that all matches should be returned.

post_refresh(*id*)

Request a refresh for column and table statistics

Parameters id : integer

Returns owner : string

The database username of the table's owner.

last_run : dict:

```
- finished_at : string/time
    The time that the run completed.
- state : string
- id : integer
- error : string
    The error message for this run, if present.
- created_at : string/time
    The time that the run was queued.
- started_at : string/time
    The time that the run started.
```

column_count : integer

The number of columns in the table.

database_id : integer

The ID of the database.

refresh_id : string

The ID of the most recent statistics refresh.

schema : string

The name of the schema containing the table.

name : string

Name of the table.

view_def : string

columns : list:

```
- useable_as_primary_key : boolean
    Whether the column may be used as an primary key to identify␣
↪table
    rows.
- 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.
- sql_type : string
    SQL type of the column.
- min_value : string
    Smallest value in the column.
- useable_as_independent_variable : boolean
    Whether the column may be used as an independent variable to␣
↪train a
```

```
    model.
- value_distribution : dict
    An object mapping distinct values in the column to the number␣
↪of times
    they appear in the column
- distinct_count : integer
    Number of distinct values in the column.
- order : integer
    Relative position of the column in the table.
- avg_value : number/float
    Average value of the column, where applicable.
- description : string
    The description of the column, as specified by the table owner
- null_count : integer
    Number of null values in the column.
- encoding : string
    The compression encoding for this columnSee: http://docs.aws.
↪amazon.com
    /redshift/latest/dg/c_Compression_encodings.html
- sample_values : list
    A sample of values from the column.
- name : string
    Name of the column.
- max_value : string
    Largest value in the column.
- stddev : number/float
    Stddev of the column, where applicable.
- coverage_count : integer
    Number of non-null 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.
```

ontology_mapping : dict

The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.

id : integer

The ID of the table.

row_count : integer

The number of rows in the table.

multipart_key : list

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.

outgoing_table_matches : list:

```
- source_table_id : integer
    Source table
```

```
- target : dict::
    - name : string
- target_id : integer
    Target ID
- target_type : string
    Target type
- job : dict::
    - match_options : dict::
        - max_matches : integer
        - threshold : string
    - state : string
        Whether the job is idle, queued, running, cancelled, or␣
↪failed.
    - runs : list::
        Information about the most recent runs of the job.
        - finished_at : string/time
            The time that the run completed.
        - state : string
        - id : integer
        - error : string
            The error message for this run, if present.
        - created_at : string/time
            The time that the run was queued.
        - started_at : string/time
            The time that the run started.
    - type : string
    - name : string
    - last_run : dict::
        - finished_at : string/time
            The time that the run completed.
        - state : string
        - id : integer
        - error : string
            The error message for this run, if present.
        - created_at : string/time
            The time that the run was queued.
        - started_at : string/time
            The time that the run started.
    - id : integer
    - hidden : boolean
        The hidden status of the object.
    - created_at : string/date-time
    - updated_at : string/date-time
```

size_mb : number/float

The size of the table in megabytes.

description : string

The description of the table, as specified by the table owner

is_view : boolean

True if this table represents a view. False if it represents a regular table.

enhancements : list:

```
- updated_at : string/time
- join_id : integer
```

```
- created_at : string/time
- type : string
```

last_refresh : string/date-time

The time of the last statistics refresh.

sortkeys : string

The column used as the Amazon Redshift sortkey.

joins : list:

```
- left_table_id : integer
- right_table_id : integer
- left_identifier : string
- left_join : boolean
- id : integer
- right_identifier : string
- on : string
- updated_at : string/time
- created_at : string/time
```

distkey : string

The column used as the Amazon Redshift distkey.

Users

class `Users` (*session*, *return_type='civis'*)

Methods

delete_api_keys(id, key_id)	Revoke the specified API key
get(id)	Show info about a user
get_api_keys(id, key_id)	Show the specified API key
list(**kwargs)	List users
list_api_keys(id, **kwargs)	Show API keys belonging to the specified user
list_me()	Show info about the logged-in user
list_me_api_accesses(**kwargs)	Show API accesses made by or on behalf of the logged-in user
patch_me(**kwargs)	Update info about the logged-in user
post_api_keys(id, expires_in, name, **kwargs)	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 **revoked_at** : string/date-time

The date and time when the key was revoked.

use_count : integer

The number of times the key has been used.

scopes : list

The scopes which the key is permissioned on.

name : string

The name of the API key.

active : boolean

True if the key has neither expired nor been revoked.

constraints : list:

```
Constraints on the abilities of the created key
- put_allowed : boolean
    Whether the constraint allows PUT requests.
- head_allowed : boolean
    Whether the constraint allows HEAD requests.
- constraint : string
    The path matcher of the constraint.
- constraint_type : string
    The type of constraint (exact/prefix/regex/verb).
- post_allowed : boolean
    Whether the constraint allows POST requests.
- patch_allowed : boolean
    Whether the constraint allows PATCH requests.
- get_allowed : boolean
    Whether the constraint allows GET requests.
- delete_allowed : boolean
    Whether the constraint allows DELETE requests.
```

id : integer

The ID of the API key.

last_used_at : string/date-time

The date and time when the key was last used.

expired : boolean

True if the key has expired.

created_at : string/date-time

The date and time when the key was created.

expires_at : string/date-time

The date and time when the key expired.

get(*id*)

Show info about a user

Parameters **id** : integer

The ID of this user.

Returns **state** : string

The state of this user.

email : string

The email of this user.

active : string

The account status of this user.

department : string

The deartment of this user.

phone : string

The phone number of this user.

city : string

The city of this user.

time_zone : string

The time zone of this user.

title : string

The title of this user.

groups : list:

```
An array of all the groups this user is in.
- id : integer
    The ID of this group.
- organization_id : integer
    The organization associated with this group.
- name : string
    The name of this group.
```

primary_group_id : integer

The ID of the primary group of this user.

initials : string

The initials of this user.

name : string

The name of this user.

github_username : string

The GitHub username of this user.

id : integer

The ID of this user.

prefers_sms_otp : string

The preference for phone authorization of this user

user : string

The username 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 **revoked_at** : string/date-time

The date and time when the key was revoked.

use_count : integer

The number of times the key has been used.

scopes : list

The scopes which the key is permissioned on.

name : string

The name of the API key.

active : boolean

True if the key has neither expired nor been revoked.

constraints : list:

```
Constraints on the abilities of the created key
- put_allowed : boolean
    Whether the constraint allows PUT requests.
- head_allowed : boolean
    Whether the constraint allows HEAD requests.
- constraint : string
    The path matcher of the constraint.
- constraint_type : string
    The type of constraint (exact/prefix/regex/verb).
- post_allowed : boolean
    Whether the constraint allows POST requests.
- patch_allowed : boolean
    Whether the constraint allows PATCH requests.
- get_allowed : boolean
    Whether the constraint allows GET requests.
- delete_allowed : boolean
    Whether the constraint allows DELETE requests.
```

id : integer

The ID of the API key.

last_used_at : string/date-time

The date and time when the key was last used.

expired : boolean

True if the key has expired.

created_at : string/date-time

The date and time when the key was created.

expires_at : string/date-time

The date and time when the key expired.

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 **groups** : list:

```
An array of all the groups this user is in.
- id : integer
    The ID of this group.
- organization_id : integer
    The organization associated with this group.
- name : string
    The name of this group.
```

primary_group_id : integer

The ID of the primary group of this user.

email : string

The email of this user.

current_sign_in_at : string/date-time

The date and time when the user's current session began.

name : string

The name of this user.

id : integer

The ID of this user.

created_at : string/date-time

The date and time when the user was created.

user : string

The username of this user.

list_api_keys(*id*, ***kwargs*)

Show API keys belonging to the specified user

Parameters **id** : string

The ID of the user or 'me'.

limit : integer, optional

Number of results to return. Defaults to its maximum of 50.

page_num : integer, optional

Page number of the results to return. Defaults to the first page, 1.

order : string, optional

The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir : string, optional

Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

iterator : bool, optional

If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns **constraint_count** : integer

The number of constraints on the created key

revoked_at : string/date-time

The date and time when the key was revoked.

use_count : integer

The number of times the key has been used.

scopes : list

The scopes which the key is permissioned on.

name : string

The name of the API key.

active : boolean

True if the key has neither expired nor been revoked.

id : integer

The ID of the API key.

last_used_at : string/date-time

The date and time when the key was last used.

expired : boolean

True if the key has expired.

created_at : string/date-time

The date and time when the key was created.

expires_at : string/date-time

The date and time when the key expired.

list_me()

Show info about the logged-in user

Returns groups : list:

```
An array of all the groups this user is in.
- id : integer
    The ID of this group.
- organization_id : integer
    The organization associated with this group.
- name : string
    The name of this group.
```

preferences : dict

This user's preferences.

username : string

This user's username.

email : string

This user's email address.

initials : string

This user's initials.

name : string

This user's name.

last_checked_announcements : string/date-time

The date and time at which the user last checked their announcements.

id : integer

The ID of this user.

feature_flags : dict

The feature flag settings for this user.

custom_branding : string

The branding of Platform for this user.

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 **path** : string

status_code : string

requested_version : integer

processing_seconds : number/float

ip : string

id : integer

api_key_id : integer

created_at : string/time

method : string

patch_me(***kwargs*)

Update info about the logged-in user

Parameters **last_checked_announcements** : string/date-time, optional

The date and time at which the user last checked their announcements.

preferences : dict, optional:

```
- project_index_archived_filter : string
    Archived filter for the projects index page.
- app_index_order_dir : string
    Oder direction for the apps index pages.
- export_index_order_field : string
    Order field for the exports index page.
- export_index_status_filter : string
    Status filter for the exports index page.
- import_index_order_dir : string
    Order direction for the imports index page.
- app_index_order_field : string
    Order field for the apps index pages.
- script_index_order_dir : string
    Order direction for the scripts index page.
- result_index_archived_filter : string
    Archived filter for the results index page.
- export_index_order_dir : string
    Order direction for the exports index page.
- preferred_server_id : integer
    ID of preferred server.
```

```
- script_index_type_filter : string
    Type filter for the scripts index page.
- result_index_order_field : string
    Order field for the results index page.
- model_index_status_filter : string
    Status filter for the models index page.
- result_index_type_filter : string
    Type filter for the results index page.
- project_detail_author_filter : string
    Author filter for projects detail pages.
- project_detail_order_dir : string
    Order direction for projects detail pages.
- script_index_author_filter : string
    Author filter for the scripts index page.
- model_index_thumbnail_view : string
    Thumbnail view for the models index page.
- export_index_type_filter : string
    Type filter for the exports index page.
- model_index_order_dir : string
    Order direction for the models index page.
- script_index_order_field : string
    Order field for the scripts index page.
- import_index_type_filter : string
    Type filter for the imports index page.
- export_index_author_filter : string
    Author filter for the exports index page.
- project_detail_type_filter : string
    Type filter for projects detail pages.
- import_index_order_field : string
    Order field for the imports index page.
- project_index_author_filter : string
    Author filter for the projects index page.
- project_index_order_field : string
    Order field for the projects index page.
- import_index_dest_filter : string
    Destination filter for the imports index page.
- project_detail_archived_filter : string
    Arhived filter for the projects detail pages.
- import_index_archived_filter : string
    Archived filter for the imports index page.
- script_index_status_filter : string
    Status filter for the scripts index page.
- import_index_status_filter : string
    Status filter for the imports index page.
- model_index_order_field : string
    Order field for the models index page.
- model_index_author_filter : string
    Author filter for the models index page.
- model_index_archived_filter : string
    Archived filter for the models index page.
- script_index_archived_filter : string
    Archived filter for the scripts index page.
- report_index_thumbnail_view : string
    Thumbnail view for the reports index page.
- project_index_order_dir : string
    Order direction for the projects index page.
- import_index_author_filter : string
    Author filter for the imports index page.
```

```
- result_index_order_dir : string
    Order direction for the results index page.
- project_detail_order_field : string
    Order field for projects detail pages.
- result_index_author_filter : string
    Author filter for the results index page.
```

Returns **groups** : list:

```
An array of all the groups this user is in.
- id : integer
    The ID of this group.
- organization_id : integer
    The organization associated with this group.
- name : string
    The name of this group.
```

preferences : dict

This user's preferences.

username : string

This user's username.

email : string

This user's email address.

initials : string

This user's initials.

name : string

This user's name.

last_checked_announcements : string/date-time

The date and time at which the user last checked their announcements.

id : integer

The ID of this user.

feature_flags : dict

The feature flag settings for this user.

custom_branding : string

The branding of Platform for this user.

post_api_keys(*id*, *expires_in*, *name*, ***kwargs*)

Create a new API key belonging to the logged-in user

Parameters **id** : string

The ID of the user or 'me'.

expires_in : integer

The number of seconds the key should last for.

name : string

The name of the API key.

constraints : list, optional:

```
Constraints on the abilities of the created key.
- put_allowed : boolean
    Whether the constraint allows PUT requests.
- head_allowed : boolean
    Whether the constraint allows HEAD requests.
- constraint : string
    The path matcher of the constraint.
- constraint_type : string
    The type of constraint (exact/prefix/regex/verb).
- post_allowed : boolean
    Whether the constraint allows POST requests.
- patch_allowed : boolean
    Whether the constraint allows PATCH requests.
- get_allowed : boolean
    Whether the constraint allows GET requests.
- delete_allowed : boolean
    Whether the constraint allows DELETE requests.
```

Returns **revoked_at** : string/date-time

The date and time when the key was revoked.

last_used_at : string/date-time

The date and time when the key was last used.

active : boolean

True if the key has neither expired nor been revoked.

constraints : list:

```
Constraints on the abilities of the created key
- put_allowed : boolean
    Whether the constraint allows PUT requests.
- head_allowed : boolean
    Whether the constraint allows HEAD requests.
- constraint : string
    The path matcher of the constraint.
- constraint_type : string
    The type of constraint (exact/prefix/regex/verb).
- post_allowed : boolean
    Whether the constraint allows POST requests.
- patch_allowed : boolean
    Whether the constraint allows PATCH requests.
- get_allowed : boolean
    Whether the constraint allows GET requests.
- delete_allowed : boolean
    Whether the constraint allows DELETE requests.
```

use_count : integer

The number of times the key has been used.

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.

id : integer

The ID of the API key.

expires_at : string/date-time

The date and time when the key expired.

expired : boolean

True if the key has expired.

scopes : list

The scopes which the key is permissioned on.

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

Index

E

F

G

I

J

L

M

P

Q

R

S

T

U