Civis Client Documentation

Release 1.16.0

Civis Analytics

Dec 14, 2021

CONTENTS

The Civis Platform 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.

API KEYS

In order to make requests to the Civis API, you will need a Civis Platform API key that is unique to you. Instructions for creating a new key are found here. API keys have a set expiration date and new keys will need to be created at least every 30 days. The API client will look for a `CIVIS_API_KEY` environmental variable to access your API key, so after creating a new API key, follow the steps below for your operating system to set up your environment.

1.1 Linux / MacOS

1. Add the following to your shell configuration file (`~/.zshrc` for MacOS or `~/.bashrc` for Linux, by default):

```
export CIVIS_API_KEY="alphaNumericApiK3y"
```

2. Source your shell configuration file (or restart your terminal).

1.2 Windows 10

1. Navigate to "Settings" -> type "environment" in search bar -> "Edit environment variables for your account". This can also be found in "System Properties" -> "Advanced" -> "Environment Variables…".

2. In the user variables section, if `CIVIS_API_KEY` already exists in the list of environment variables, click on it and press "Edit…". Otherwise, click "New..".

3. Enter CIVIS_API_KEY as the "Variable name".

4. Enter your API key as the "Variable value". Your API key should look like a long string of letters and numbers.

INSTALLATION

After creating an API key and setting the `CIVIS_API_KEY` environmental variable, install the Python package `civis` with the recommended method via `pip`:

```
pip install civis
```

Alternatively, if you are interested in the latest functionality not yet released through `pip`, you may clone the code from GitHub and build from source:

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

You can test your installation by running

```
import civis
client = civis.APIClient()
print(client.users.list_me()['username'])
```

If `civis` was installed correctly, this will print your Civis Platform username.

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

```
pip install pandas
```

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

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

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

```
pip install civisml-extensions
pip install glmnet
pip install muffnn
```

USER GUIDE

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

RETRIES

The API client will automatically retry for certain API error responses.

If the error is one of [413, 429, 503] and the API client is told how long it needs to wait before it's safe to retry (this is always the case with 429s, which are rate limit errors), then the client will wait the specified amount of time before retrying the request.

If the error is one of [429, 502, 503, 504] and the request is not a `patch*` or `post*` method, then the API client will retry the request several times, with an exponential delay, to see if it will succeed. If the request is of type `post*` it will retry with the same parameters for error codes [429, 503].

CLIENT API REFERENCE

5.1 User Guide

5.1.1 Getting Started

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

```
>>> import civis
```

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

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

5.1.2 Civis Futures

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

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

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

```
>>> import civis
>>> import concurrent.futures
```

(continues on next page)

```
>>>
>>> client = civis.APIClient()
>>>
>>> # Create a container script. This is just a simple example. Futures can
>>> # also be used with SQL queries, imports, etc.
>>> response_script = client.scripts.post_containers(
...     required_resources={'cpu': 512, 'memory': 1024},
...     docker_command="echo 'HELLO WORLD'",
...     docker_image_name='civisanalytics/datascience-python')
>>> script_id = response_script.id
>>>
>>> # Create a run in order to execute the script.
>>> response_run = client.scripts.post_containers_runs(script_id)
>>> run_id = response_run.id
>>>
>>> # Create a future to represent the result of the run.
>>> future = civis.futures.CivisFuture(
...     client.scripts.get_containers_runs, (script_id, run_id))
>>>
>>> # You can then have your code block and wait for the future to be done as
>>> # follows. Note that this does not raise an exception on error like
>>> # `future.result()`.
>>> concurrent.futures.wait([future])
>>>
>>> # Alternatively, you can call `future.result()` to block and get the
>>> # status of the run once it finishes. If the run is already completed, the
>>> # result will be returned immediately.
>>> result = future.result()
>>>
>>> # Alternatively, one can start a run and get a future for it with the helper
>>> # function `civis.utils.run_job`:
>>> future2 = civis.utils.run_job(script_id)
>>> future2.result()
```

5.1.3 Working Directly with the Client

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

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

Note: Creating an instance of *APIClient* makes an HTTP request to determine the functions to attach to the object. You must have an API key and internet connection to create an *APIClient* object.

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'}
```

For a complete list of the API endpoints and their methods, check out *API Resources*.

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

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

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

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

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

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

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

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

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

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

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

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

5.1.4 API Response Types and Functions

Many API requests via an *APIClient* instance return an iterable of *civis.response.Response* objects. For end-points that support pagination when the *iterator* kwarg is specified, a *civis.response.PaginatedResponse* object is returned. To facilitate working with *civis.response.Response* objects, the helper functions *civis.find()* and *civis.find_one()* are defined.

5.1.5 Testing Your Code

Once you've written code that uses *APIClient*, you've got to test it. Because you want a testing environment not dependent upon an API key or an internet connection, you will employ the mocking technique.

To this end, `civis.tests.create_client_mock()` will create a mock object that looks like an API client object. This mock object is configured to error if any method calls have non-existent / misspelled parameters.

Suppose this function is in your code:

```python
def get_timestamps_from_table(..., client=None, ...):
    ...
    client = client if not client else civis.APIClient()
    ...
    df = civis.io.read_civis_sql(
        ...,
        client=client,
        ...,
    )
    ...
    return ...
```

Whatever function you define, it needs to have a `client` argument. If it's not provided, an actual API client object will be created. Throughout this function, the `client` object has to be used to interact with the Civis API. It is through this argument that you as a developer can pass in a custom API client object.

When you're testing your functions in your test suite, you might have code like this:

```python
from civis.tests import create_client_mock

from <your-package> import get_timestamps_from_table

def test_get_timestamps_from_table():
    mock_client = create_client_mock()

    mock_client.scripts.get_sql_runs.return_value = {
        "output": [
            {
                "path": ...
                "file_id": ...
            }
        ]
    }
    actual_timestamps = get_timestamps_from_table(
        ...
        client=mock_client,
        ...
```

(continues on next page)

```
    )

    expected_timestamps = ...

    # Run assertion tests as necessary
    assert actual_timestamps == expected_timestamps
```

Once you've created a mock client object, you have to define its behavior based on expected API calls from the function you've defined. Also, be sure to use `mock_client` so you don't actually have to process an actual API call in your test.

5.2 Data Import and Export

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

5.2.1 Tables

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

civis_to_csv(filename, sql, database[, ...])	Export data from Civis to a local CSV file.
civis_to_multifile_csv(sql, database[, ...])	Unload the result of SQL query and return presigned urls.
civis_file_to_table(file_id, database, table)	Upload the contents of one or more Civis files to a Civis table.
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.
export_to_civis_file(sql, database[, ...])	Store results of a query to a Civis file
split_schema_tablename(table)	Split a Redshift 'schema.tablename' string

civis.io.civis_to_csv

civis.io.**civis_to_csv**(*filename*, *sql*, *database*, *job_name=None*, *api_key=None*, *client=None*,
credential_id=None, *include_header=True*, *compression='none'*, *delimiter=','*,
unquoted=False, *archive=False*, *hidden=True*, *polling_interval=None*)
Export data from Civis to a local CSV file.

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

> **Parameters**
>
> > **filename** [str] Download exported data into this file.
> >
> > **sql** [str] The SQL select string to be executed.
> >
> > **database** [str or int] Export data from this database. Can be the database name or ID.

job_name [str, optional] A name to give the job. If omitted, a random job name will be used.

api_key [DEPRECATED str, optional] Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

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

credential_id [str or int, optional] The ID of the database credential. If `None`, the default credential will be used.

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

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

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

unquoted: bool, optional Whether or not to quote fields. Default: `False`.

polling_interval [int or float, optional] Number of seconds to wait between checks for query completion.

archive [bool, optional (deprecated)] If `True`, archive the import job as soon as it completes.

hidden [bool, optional] If `True` (the default), this job will not appear in the Civis UI.

Returns

results [`CivisFuture`] A *CivisFuture* object.

See also:

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

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

`civis.io.export_to_civis_file` Store a SQL query's results in a Civis file

Examples

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

civis.io.civis_to_multifile_csv

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

Unload the result of SQL query and return presigned urls.

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

Parameters

sql [str] The SQL select string to be executed.

database [str or int] Execute the query against this database. Can be the database name or ID.

job_name [str, optional] A name to give the job. If omitted, a random job name will be used.

api_key [DEPRECATED str, optional] Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

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

credential_id [str or int, optional] The database credential ID. If `None`, the default credential will be used.

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

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

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

max_file_size: int, optional Maximum number of Megabytes each created file will be.

unquoted: bool, optional Whether or not to quote fields. Default: `False`.

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

polling_interval [int or float, optional] Number of seconds to wait between checks for query completion.

hidden [bool, optional] If `True` (the default), this job will not appear in the Civis UI.

Returns

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

'query': str The query.

'header': list of str The columns from the query.

'entries': list of dict Each dict has the following keys:

'id': int File ID

'name': str Filename

'size': int File size in bytes

'url': str Unsigned S3 URL ('s3://…')

'url_signed': str Signed S3 URL ('https://…')

'unquoted': bool Whether the cells are quoted.

'compression': str Type of compression used.

'delimiter': str Delimiter that separates the cells.

See also:

`civis.APIClient.scripts.post_sql`

Examples

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

civis.io.civis_file_to_table

civis.io.**civis_file_to_table**(*file_id*, *database*, *table*, *client=None*, *max_errors=None*, *existing_table_rows='fail'*, *diststyle=None*, *distkey=None*, *sortkey1=None*, *sortkey2=None*, *table_columns=None*, *primary_keys=None*, *last_modified_keys=None*, *escaped=False*, *execution='immediate'*, *delimiter=None*, *headers=None*, *credential_id=None*, *polling_interval=None*, *hidden=True*)

Upload the contents of one or more Civis files to a Civis table. All provided files will be loaded as an atomic unit in parallel, and should share the same columns in the same order, and be in the same format.

Note: Civis files must be in a CSV-like delimiter separated format.

Parameters

file_id [int or list[int]] Civis file ID or a list of Civis file IDs.

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'`.

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

max_errors [int, optional] The maximum number of rows with errors to remove from the import before failing. If multiple files are provided, this limit applies across all files combined.

existing_table_rows [str, optional] The behaviour if a table with the requested name already exists. One of `'fail'`, `'truncate'`, `'append'`, `'drop'`, or `'upsert'`. Defaults to `'fail'`.

diststyle [str, optional] The distribution style for the table. One of `'even'`, `'all'` or `'key'`.

distkey [str, optional] The column to use as the distkey for the table.

sortkey1 [str, optional] The column to use as the sortkey for the table.

sortkey2 [str, optional] The second column in a compound sortkey for the table.

table_columns [list[Dict[str, str]], optional] A list of dictionaries, ordered so each dictionary corresponds to a column in the order that it appears in the source file. Each dict should have a key "name" that corresponds to the column name in the destination table, and a key "sql_type" corresponding to the intended column data type in the destination table. The "sql_type" key is not required when appending to an existing table. The table_columns parameter is required if the table does not exist, the table is being dropped, or the columns in the

source file do not appear in the same order as in the destination table. Example: `[{"name": "foo", "sql_type": "INT"}, {"name": "bar", "sql_type": "VARCHAR"}]`

primary_keys: list[str], optional A list of the primary key column(s) of the destination table that uniquely identify a record. These columns must not contain null values. If existing_table_rows is "upsert", this field is required. Note that this is true regardless of whether the destination database itself requires a primary key.

last_modified_keys: list[str], optional A list of the columns indicating a record has been updated. If existing_table_rows is "upsert", this field is required.

escaped: bool, optional A boolean value indicating whether or not the source file(s) escape quotes with a backslash. Defaults to false.

execution: string, optional, default "immediate" One of "delayed" or "immediate". If "immediate", refresh column statistics as part of the run. If "delayed", flag the table for a deferred statistics update; column statistics may not be available for up to 24 hours. In addition, if existing_table_rows is "upsert", delayed executions move data from staging table to final table after a brief delay, in order to accommodate multiple concurrent imports to the same destination table.

delimiter [string, optional] The column delimiter. One of `','`, `'\t'` or `'|'`. If not provided, will attempt to auto-detect.

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.

hidden [bool, optional] If `True` (the default), this job will not appear in the Civis UI.

Returns

results [*CivisFuture*] A *CivisFuture* object.

Raises

CivisImportError If multiple files are given and determined to be incompatible for import. This may be the case if their columns have different types, their delimiters are different, headers are present in some but not others, or compressions do not match.

Examples

```
>>> file_id = 100
>>> fut = civis.io.civis_file_to_table(file_id,
...                                    'my-database',
...                                    'scratch.my_data')
>>> fut.result()
```

civis.io.csv_to_civis

civis.io.**csv_to_civis**(*filename*, *database*, *table*, *api_key=None*, *client=None*, *max_errors=None*, *existing_table_rows='fail'*, *diststyle=None*, *distkey=None*, *sortkey1=None*, *sortkey2=None*, *table_columns=None*, *delimiter=','*, *headers=None*, *primary_keys=None*, *last_modified_keys=None*, *escaped=False*, *execution='immediate'*, *credential_id=None*, *polling_interval=None*, *archive=False*, *hidden=True*)

Upload the contents of a local CSV file to Civis.

> **Parameters**
>
> > **filename** [str] Upload the contents of this file.
> >
> > **database** [str or int] Upload data into this database. Can be the database name or ID.
> >
> > **table** [str] The schema and table you want to upload to. E.g., `'scratch.table'`.
> >
> > **api_key** [DEPRECATED str, optional] Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.
> >
> > **client** [`civis.APIClient`, optional] If not provided, an `civis.APIClient` object will be created from the `CIVIS_API_KEY`.
> >
> > **max_errors** [int, optional] The maximum number of rows with errors to remove from the import before failing.
> >
> > **existing_table_rows** [str, optional] The behaviour if a table with the requested name already exists. One of `'fail'`, `'truncate'`, `'append'`, `'drop'`, or `'upsert'`. Defaults to `'fail'`.
> >
> > **diststyle** [str, optional] The distribution style for the table. One of `'even'`, `'all'` or `'key'`.
> >
> > **distkey** [str, optional] The column to use as the distkey for the table.
> >
> > **sortkey1** [str, optional] The column to use as the sortkey for the table.
> >
> > **sortkey2** [str, optional] The second column in a compound sortkey for the table.
> >
> > **table_columns** [list[Dict[str, str]], optional] A list of dictionaries, ordered so each dictionary corresponds to a column in the order that it appears in the source file. Each dict should have a key "name" that corresponds to the column name in the destination table, and a key "sql_type" corresponding to the intended column data type in the destination table. The "sql_type" key is not required when appending to an existing table. The table_columns parameter is required if the table does not exist, the table is being dropped, or the columns in the source file do not appear in the same order as in the destination table. Example: [{"name": "foo", "sql_type": "INT"}, {"name": "bar", "sql_type": "VARCHAR"}]
> >
> > **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.
> >
> > **primary_keys: list[str], optional** A list of the primary key column(s) of the destination table that uniquely identify a record. These columns must not contain null values. If existing_table_rows is "upsert", this field is required. Note that this is true regardless of whether the destination database itself requires a primary key.
> >
> > **last_modified_keys: list[str], optional** A list of the columns indicating a record has been updated. If existing_table_rows is "upsert", this field is required.
> >
> > **escaped: bool, optional** A boolean value indicating whether or not the source file has quotes escaped with a backslash. Defaults to false.

execution: string, optional, default "immediate" One of "delayed" or "immediate". If "immediate", refresh column statistics as part of the run. If "delayed", flag the table for a deferred statistics update; column statistics may not be available for up to 24 hours. In addition, if existing_table_rows is "upsert", delayed executions move data from staging table to final table after a brief delay, in order to accommodate multiple concurrent imports to the same destination table.

credential_id [str or int, optional] The ID of the database credential. If `None`, the default credential will be used.

polling_interval [int or float, optional] Number of seconds to wait between checks for job completion.

archive [bool, optional (deprecated)] If `True`, archive the import job as soon as it completes.

hidden [bool, optional] If `True` (the default), this job will not appear in the Civis UI.

Returns

results [*CivisFuture*] A *CivisFuture* object.

Notes

This reads the contents of *filename* into memory.

Examples

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

civis.io.dataframe_to_civis

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

Upload a *pandas DataFrame* into a Civis table.

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

Parameters

df [*pandas.DataFrame*] The *DataFrame* to upload to Civis.

database [str or int] Upload data into this database. Can be the database name or ID.

table [str] The schema and table you want to upload to. E.g., `'scratch.table'`. Schemas or tablenames with periods must be double quoted, e.g. `'scratch."my.table"'`.

api_key [DEPRECATED str, optional] Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

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

max_errors [int, optional] The maximum number of rows with errors to remove from the import before failing.

existing_table_rows [str, optional] The behaviour if a table with the requested name already exists. One of `'fail'`, `'truncate'`, `'append'`, `'drop'`, or `'upsert'`. Defaults to `'fail'`.

diststyle [str, optional] The distribution style for the table. One of `'even'`, `'all'` or `'key'`.

distkey [str, optional] The column to use as the distkey for the table.

sortkey1 [str, optional] The column to use as the sortkey for the table.

sortkey2 [str, optional] The second column in a compound sortkey for the table.

table_columns [list[Dict[str, str]], optional] A list of dictionaries, ordered so each dictionary corresponds to a column in the order that it appears in the source file. Each dict should have a key "name" that corresponds to the column name in the destination table, and a key "sql_type" corresponding to the intended column data type in the destination table. The "sql_type" key is not required when appending to an existing table. The table_columns parameter is required if the table does not exist, the table is being dropped, or the columns in the source file do not appear in the same order as in the destination table. Example: `[{"name": "foo", "sql_type": "INT"}, {"name": "bar", "sql_type": "VARCHAR"}]`

headers [bool, optional [DEPRECATED]] 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.

This parameter has no effect in versions >= 1.11 and will be removed in v2.0. Tables will always be written with column names read from the DataFrame. Use the *header* parameter (which will be passed directly to `to_csv()`) to modify the column names in the Civis Table.

credential_id [str or int, optional] The ID of the database credential. If `None`, the default credential will be used.

primary_keys: list[str], optional A list of the primary key column(s) of the destination table that uniquely identify a record. These columns must not contain null values. If existing_table_rows is "upsert", this field is required. Note that this is true regardless of whether the destination database itself requires a primary key.

last_modified_keys: list[str], optional A list of the columns indicating a record has been updated. If existing_table_rows is "upsert", this field is required.

escaped: bool, optional A boolean value indicating whether or not the source file has quotes escaped with a backslash. Defaults to false.

execution: string, optional, default "immediate" One of "delayed" or "immediate". If "immediate", refresh column statistics as part of the run. If "delayed", flag the table for a deferred statistics update; column statistics may not be available for up to 24 hours. In addition, if existing_table_rows is "upsert", delayed executions move data from staging table to final table after a brief delay, in order to accommodate multiple concurrent imports to the same destination table.

polling_interval [int or float, optional] Number of seconds to wait between checks for job completion.

archive [bool, optional (deprecated)] If `True`, archive the import job as soon as it completes.

hidden [bool, optional] If `True` (the default), this job will not appear in the Civis UI.

**kwargs [kwargs] Extra keyword arguments will be passed to `pandas.DataFrame.to_csv()`.

Returns

> **fut** [*CivisFuture*] A *CivisFuture* object.

See also:

`to_csv()`

Examples

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

civis.io.read_civis

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

Read data from a Civis table.

Parameters

table [str] Name of table, including schema, in the database. E.g. `'my_schema.my_table'`. Schemas or tablenames with periods must be double quoted, e.g. `'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()`.

encoding [str, optional] If `use_pandas` is `True`, this parameter is passed to the `encoding` kwarg of `pandas.read_csv()`. If `use_pandas` is `False`, and if this parameter isn't provided, then the UTF-8 encoding is assumed. In case you encounter a `UnicodeDecodeError`, consider choosing an encoding suitable for your data; see the list of standard encodings.

job_name [str, optional] A name to give the job. If omitted, a random job name will be used.

api_key [DEPRECATED str, optional] Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

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

credential_id [str or int, optional] The database credential ID. If `None`, the default credential will be used.

polling_interval [int or float, optional] Number of seconds to wait between checks for query completion.

archive [bool, optional (deprecated)] If `True`, archive the import job as soon as it completes.

hidden [bool, optional] If `True` (the default), this job will not appear in the Civis UI.

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

Returns

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

Raises

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

EmptyResultError If the table is empty.

See also:

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

civis.io.civis_to_csv Write directly to csv.

civis.io.export_to_civis_file Store a SQL query's results in a Civis file

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*, *encoding=None*, *job_name=None*, *api_key=None*, *client=None*, *credential_id=None*, *polling_interval=None*, *archive=False*, *hidden=True*, ***kwargs*)
Read data from Civis using a custom SQL string.

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

Parameters

sql [str] 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()`.

encoding [str, optional] If `use_pandas` is `True`, this parameter is passed to the `encoding` kwarg of `pandas.read_csv()`. If `use_pandas` is `False`, and if this parameter isn't provided, then the UTF-8 encoding is assumed. In case you encounter a `UnicodeDecodeError`, consider choosing an encoding suitable for your data; see the list of standard encodings.

job_name [str, optional] A name to give the job. If omitted, a random job name will be used.

api_key [DEPRECATED str, optional] Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

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

credential_id [str or int, optional] The database credential ID. If `None`, the default credential will be used.

polling_interval [int or float, optional] Number of seconds to wait between checks for query completion.

archive [bool, optional (deprecated)] If `True`, archive the import job as soon as it completes.

hidden [bool, optional] If `True` (the default), this job will not appear in the Civis UI.

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

Returns

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

Raises

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

EmptyResultError If no rows were returned as a result of the query.

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]
```

civis.io.export_to_civis_file

civis.io.**export_to_civis_file**(*sql*, *database*, *job_name=None*, *client=None*, *credential_id=None*, *polling_interval=None*, *hidden=True*, *csv_settings=None*)

> Store results of a query to a Civis file
>
> > **Parameters**
> >
> > > **sql** [str] The SQL select string to be executed.
> > >
> > > **database** [str or int] Execute the query against this database. Can be the database name or ID.
> > >
> > > **job_name** [str, optional] A name to give the job. If omitted, a random job name will be used.
> > >
> > > **client** [*civis.APIClient*, optional] If not provided, an *civis.APIClient* object will be created from the CIVIS_API_KEY.
> > >
> > > **credential_id** [str or int, optional] The database credential ID. If None, the default credential will be used.
> > >
> > > **polling_interval** [int or float, optional] Number of seconds to wait between checks for query completion.
> > >
> > > **hidden** [bool, optional] If True (the default), this job will not appear in the Civis UI.
> > >
> > > **csv_settings** [dict, optional] A dictionary of csv_settings to pass to civis.APIClient. scripts.post_sql().
> >
> > **Returns**
> >
> > > **fut** [*CivisFuture*] A future which returns the response from civis.APIClient.scripts. get_sql_runs() after the sql query has completed and the result has been stored as a Civis file.
>
> **See also:**
>
> *civis.io.read_civis* Read directly into memory without SQL.
>
> *civis.io.read_civis_sql* Read results of a SQL query into memory.
>
> *civis.io.civis_to_csv* Write directly to a CSV file.
>
> *civis.io.civis_file_to_table* Upload a Civis file to a Civis table

Examples

```
>>> sql = "SELECT * FROM schema.table"
>>> fut = export_to_civis_file(sql, "my_database")
>>> file_id = fut.result()['output'][0]["file_id"]
```

civis.io.split_schema_tablename

civis.io.**split_schema_tablename**(*table*)

Split a Redshift 'schema.tablename' string

Remember that special characters (such as '.') can only be included in a schema or table name if delimited by double-quotes.

Parameters

table: str Either a Redshift schema and table name combined with a ".", or else a single table name.

Returns

schema, tablename A 2-tuple of strings. The `schema` may be None if the input is only a table name, but the `tablename` will always be filled.

Raises

ValueError If the input `table` is not separable into a schema and table name.

5.2.2 Files

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

civis_to_file(file_id, buf[, api_key, client])	Download a file from Civis.
dataframe_to_file(df[, name, expires_at, client])	Store a `DataFrame` as a CSV in Civis Platform
file_id_from_run_output(name, job_id, run_id)	Find the file ID of a File run output with the name "name"
file_to_civis(buf[, name, api_key, client])	Upload a file to Civis.
file_to_dataframe(file_id[, compression, client])	Load a `DataFrame` from a CSV stored in a Civis File
file_to_json(file_id[, client])	Restore JSON stored in a Civis File
json_to_file(obj[, name, expires_at, client])	Store a JSON-serializable object in a Civis File

civis.io.civis_to_file

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

Download a file from Civis.

Parameters

file_id [int] The Civis file ID.

buf [file-like object or str] A buffer or path specifying where to write the contents of the Civis file. Strings will be treated as paths to local files to open.

api_key [DEPRECATED str, optional] Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

> **client** [`civis.APIClient`, optional] If not provided, an `civis.APIClient` object will be created from the `CIVIS_API_KEY`.

> **Returns**

>> **None**

Examples

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

civis.io.dataframe_to_file

civis.io.**dataframe_to_file**(*df*, *name='data.csv'*, *expires_at='DEFAULT'*, *client=None*, ***to_csv_kws*)

> Store a `DataFrame` as a CSV in Civis Platform

> **Parameters**

>> **df** [`DataFrame`] The table to upload.

>> **name** [str, optional] The name of the Civis File

>> **expires_at** [str, 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 `None`. To specify a date and time, format it by the ISO 8601 standard, e.g., `"2020-12-31"`, `"2020-12-31T23:03:40Z"`, and what the `isoformat()` method returns from a `datetime.date` or `datetime.datetime` object.

>> **client** [`civis.APIClient`, optional] If not provided, an `civis.APIClient` object will be created from the `CIVIS_API_KEY`.

>> ****to_csv_kws** Additional keyword parameters will be passed directly to `to_csv()`.

> **Returns**

>> **file_id** [int] The integer ID of the new Civis File object

> **See also:**

> *file_to_civis()*

> `to_csv()`

civis.io.file_id_from_run_output

civis.io.**file_id_from_run_output**(*name*, *job_id*, *run_id*, *regex=False*, *client=None*)
> Find the file ID of a File run output with the name "name"

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

> > **Parameters**

> > > **name** [str] The "name" field of the run output you wish to retrieve

> > > **job_id** [int]

> > > **run_id** [int]

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

> > > **client** [*civis.APIClient*, optional] If not provided, an *civis.APIClient* object will be created from the CIVIS_API_KEY.

> > **Returns**

> > > **file_id** [int] The ID of a Civis File with name matching `name`

> > **Raises**

> > > **IOError** If the provided job ID and run ID combination can't be found

> > > **FileNotFoundError** If the run exists, but `name` isn't in its run outputs

> **See also:**

> **APIClient.scripts.list_containers.runs_outputs**

civis.io.file_to_civis

civis.io.**file_to_civis**(*buf*, *name=None*, *api_key=None*, *client=None*, ***kwargs*)
> Upload a file to Civis.

> > **Parameters**

> > > **buf** [file-like object or str] Either a file-like object for a buffer or a string for a local file path. Note that if a file-like object is provided and it's not an `io.BufferedReader` or `io.TextIoWrapper` object, the current implementation requires extra disk space (which could be an issue if your file is large).

> > > **name** [str, optional] The name you wish to give the file. If not given, it will be inferred from the basename of `buf` (if `buf` is a string for a file path) or `buf.name` (if `buf` is a file-like object).

> > > **api_key** [DEPRECATED str, optional] Your Civis API key. If not given, the CIVIS_API_KEY environment variable will be used.

> > > **client** [*civis.APIClient*, optional] If not provided, an *civis.APIClient* object will be created from the CIVIS_API_KEY.

> > > ****kwargs** [kwargs] Extra keyword arguments will be passed to the file creation endpoint. See *post()*. In particular, `expires_at` can be specified for the date and time the file will expire. If not specified, the file will expire in 30 days. To keep a file indefinitely, specify `None`. To specify a date and time, format it by the ISO 8601 standard, e.g.,

---

"2020-12-31", "2020-12-31T23:03:40Z", and what the isoformat() method returns from a datetime.date or datetime.datetime object.

Returns

> **file_id** [int] The new Civis file ID.

Raises

> **TypeError** If name is not provided and cannot be inferred from buf

Notes

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

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

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

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

Examples

```
>>> # Upload file at a given path on the local filesystem.
>>> file_id = file_to_civis("my_data.csv", 'my_data')
>>> # If not given, ``name`` will be the basename of the given file path.
>>> file_id = file_to_civis("foo/bar/data.csv")  # ``name`` is 'data.csv'
>>> # Upload file which expires in 30 days
>>> with open("my_data.csv", "r") as f:
...     file_id = file_to_civis(f, 'my_data')
>>> # Upload file which never expires
>>> with open("my_data.csv", "r") as f:
...     file_id = file_to_civis(f, 'my_data', expires_at=None)
```

civis.io.file_to_dataframe

civis.io.**file_to_dataframe**(*file_id*, *compression='infer'*, *client=None*, ***read_kwargs*)
    Load a DataFrame from a CSV stored in a Civis File

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

    Parameters

    > **file_id** [int] ID of a Civis File which contains a CSV
    >
    > **compression** [str, optional] If "infer", set the compression argument of pandas.read_csv based on the file extension of the name of the Civis File. Otherwise pass this argument to pandas.read_csv.
    >
    > **client** [*civis.APIClient*, optional] If not provided, an *civis.APIClient* object will be created from the CIVIS_API_KEY.
    >
    > ****read_kwargs** Additional arguments will be passed directly to read_csv().

    Returns

    > **DataFrame** containing the contents of the CSV

Raises

ImportError If pandas is not available

See also:

`pandas.read_csv`

civis.io.file_to_json

civis.io.**file_to_json**(*file_id*, *client=None*, ***json_kwargs*)
Restore JSON stored in a Civis File

Parameters

file_id [int] ID of a JSON-formatted Civis File

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

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

Returns

The object extracted from the JSON-formatted file

See also:

`civis_to_file()`

`json.load()`

civis.io.json_to_file

civis.io.**json_to_file**(*obj*, *name='file.json'*, *expires_at='DEFAULT'*, *client=None*, ***json_kwargs*)
Store a JSON-serializable object in a Civis File

Parameters

obj The object to be JSON-serialized and stored in a Civis File

name [str, optional] The name of the Civis File

expires_at [str, 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 `None`. To specify a date and time, format it by the ISO 8601 standard, e.g., `"2020-12-31"`, `"2020-12-31T23:03:40Z"`, and what the `isoformat()` method returns from a `datetime.date` or `datetime.datetime` object.

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

**json_kwargs Additional keyword arguments will be passed directly to `json.dump()`.

Returns

file_id [int] The integer ID of the new Civis File object

See also:

`file_to_civis()`

`json.dump()`

5.2.3 Databases

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

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*,
client=None, *source_credential_id=None*, *dest_credential_id=None*,
polling_interval=None, ***advanced_options*)

    Transfer a table from one location to another.

        Parameters

            source_db [str or int] The name of the database where the source table is located. Optionally,
could be the database ID.

            dest_db [str or int] The name of the database where the table will be transfered. Optionally,
could be the database ID.

            source_table [str] Full name of the table to transfer, e.g., `'schema.table'`.

            dest_table [str] Full name of the table in the destination database, e.g., `'schema.table'`.

            job_name [str, optional] A name to give the job. If omitted, a random job name will be used.

            api_key [DEPRECATED str, optional] Your Civis API key. If not given, the `CIVIS_API_KEY`
environment variable will be used.

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

            source_credential_id [str or int, optional] Optional credential ID for the source database. If
`None`, the default credential will be used.

            dest_credential_id [str or int, optional] Optional credential ID for the destination database. If
`None`, the default credential will be used.

            polling_interval [int or float, optional] Number of seconds to wait between checks for job completion.

            **advanced_options [kwargs] Extra keyword arguments will be passed to the import sync job.
See *post_syncs()*.

        Returns

            results [*CivisFuture*] A *CivisFuture* object.

Examples

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

civis.io.query_civis

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

Execute a SQL statement as a Civis query.

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

> **Parameters**
>
> > **sql** [str] The SQL statement to execute.
> >
> > **database** [str or int] The name or ID of the database.
> >
> > **api_key** [DEPRECATED str, optional] Your Civis API key. If not given, the CIVIS_API_KEY environment variable will be used.
> >
> > **client** [*civis.APIClient*, optional] If not provided, an *civis.APIClient* object will be created from the CIVIS_API_KEY.
> >
> > **credential_id** [str or int, optional] The ID of the database credential. If None, the default credential will be used.
> >
> > **preview_rows** [int, optional] The maximum number of rows to return. No more than 100 rows can be returned at once.
> >
> > **polling_interval** [int or float, optional] Number of seconds to wait between checks for query completion.
> >
> > **hidden** [bool, optional] If True (the default), this job will not appear in the Civis UI.
>
> **Returns**
>
> > **results** [*CivisFuture*] A *CivisFuture* object.

Examples

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

5.3 Machine Learning

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

5.3.1 Optional Dependencies

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

- pandas

- scikit-learn

- glmnet

- feather-format

- civisml-extensions

- muffnn

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

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

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

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

5.3.2 Define Your Model

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

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

If you have already trained a scikit-learn model outside of Civis Platform, you can register it with Civis Platform as a CivisML model so that you can score it using CivisML. Read *Registering Models Trained Outside of Civis* for how to do this.

Pre-Defined Models

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

Name	Model Type	Algorithm	Altered Defaults
sparse_logistic	classifica-tion	LogisticRegression	`C=499999950, tol=1e-08`
gradient_boosting_classifier	classifica-tion	GradientBoostingClassi-fier	`n_estimators=500,` `max_depth=2`
random_forest_classifier	classifica-tion	RandomForestClassifier	`n_estimators=500,` `max_depth=7`
extra_trees_classifier	classifica-tion	ExtraTreesClassifier	`n_estimators=500,` `max_depth=7`
multi-layer_perceptron_classifier	classifica-tion	muffnn.MLPClassifier	
stacking_classifier	classifica-tion	civism-lext.StackedClassifier	
sparse_linear_regressor	regression	LinearRegression	
sparse_ridge_regressor	regression	Ridge	
gradient_boosting_regressor	regression	GradientBoostingRegres-sor	`n_estimators=500,` `max_depth=2`
random_forest_regressor	regression	RandomForestRegressor	`n_estimators=500,` `max_depth=7`
extra_trees_regressor	regression	ExtraTreesRegressor	`n_estimators=500,` `max_depth=7`
multi-layer_perceptron_regressor	regression	muffnn.MLPRegressor	
stacking_regressor	regression	civism-lext.StackedRegressor	

The "stacking_classifier" model stacks the "gradient_boosting_classifier", and "random_forest_classifier" pre-defined models together with a `glmnet.LogitNet(alpha=0, n_splits=4, max_iter=10000, tol=1e-5,` `scoring='log_loss')`. The models are combined using a Pipeline containing a Normalizer step, followed by LogisticRegressionCV with `penalty='l2'` and `tol=1e-08`. The "stacking_regressor" works similarly, stacking together the "gradient_boosting_regressor" and "random_forest_regressor" models and a `glmnet.` `ElasticNet(alpha=0, n_splits=4, max_iter=10000, tol=1e-5, scoring='r2')`, combining them using NonNegativeLinearRegression. The estimators that are being stacked have the same names as the associated pre-defined models, and the meta-estimator steps are named "meta-estimator". Note that although default parameters are provided for multilayer perceptron models, it is highly recommended that multilayer perceptrons be run using hyperband.

Custom Models

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

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

Custom ETL

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

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

Hyperparameter Tuning

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

Hyperband is an efficient approach to hyperparameter optimization, and *recommended over grid search where possible*. CivisML will perform hyperband optimization for a pre-defined model if you pass the string `'hyperband'` to `cross_validation_parameters`. Hyperband is currently only supported for the following models: `gradient_boosting_classifier`, `random_forest_classifier`, `extra_trees_classifier`, `multilayer_perceptron_classifier`, `stacking_classifier`, `gradient_boosting_regressor`, `random_forest_regressor`, `extra_trees_regressor`, `multilayer_perceptron_regressor`, and `stacking_regressor`. Although hyperband is supported for stacking models, stacking itself is a kind of model tuning, and the combination of stacking and hyperband is likely too computationally intensive to be useful in many cases.

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

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

Models	Cost Parameter	Hyperband Distributions

gradient_boosting_classifier
gradient_boosting_regressor
GBT step in stacking_classifier
GBT step in stacking_regressor
	`n_estimators`
	`min = 100,`
	`max = 1000`
		`max_depth:  randint(low=1,`
		`high=5)`
		`max_features:  [None,`
		`'sqrt', 'log2', 0.5, 0.3,`
		`0.1, 0.05, 0.01]`
		`learning_rate:`
		`truncexpon(b=5, loc=.0003,`
		`scale=1./167.)`

random_forest_classifier
random_forest_regressor
extra_trees_classifier
extra_trees_regressor
RF step in stacking_classifier
RF step in stacking_regressor
	`n_estimators`
	`min = 100,`
	`max = 1000`
		`criterion:  ['gini',`
		`'entropy']`
		`max_features:`
		`truncexpon(b=10., loc=.01,`
		`scale=1./10.11)`
		`max_depth:  [1, 2, 3, 4, 6,`
		`10]`

multilayer_perceptron_classifier
multilayer_perceptron_regressor
	`n_epochs`
	`min = 5,`
	`max = 50`
		`keep_prob:  uniform()`
		`hidden_units:  [(), (16,),`
		`(32,), (64,), (64, 64),`
		`(64, 64, 64),`
		`    (128,), (128, 128),`
		`    (128, 128, 128),`
		`    (256,),`
		`    (256, 256), (256, 256,`
		`    256), (512, 256, 128,`
		`    64),`
		`    (1024, 512, 256, 128)]`
		`learning_rate:  [1e-2,`
		`2e-2, 5e-2, 8e-2, 1e-3,`
		`2e-3, 5e-3, 8e-3, 1e-4]`

The truncated exponential distribution for the gradient boosting classifier and regressor was chosen to skew the distribution toward small values, ranging between .0003 and .03, with a mean close to .006. Similarly, the truncated exponential distribution for the random forest and extra trees models skews toward small values, ranging between .01 and 1, and with a mean close to .1.

Custom Dependencies

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

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

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

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

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

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

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

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

Note, installing private dependencies with submodules is not supported.

CivisML Versions

By default, CivisML uses its latest version in production. If you would like a specific version (e.g., for a production pipeline where pinning the CivisML version is desirable), *ModelPipeline* (both its constructor and the class method *civis.ml.ModelPipeline.register_pretrained_model()*) has the optional parameter `civisml_version` that accepts a string, e.g., `'v2.3'` for CivisML v2.3. Please see here for the list of CivisML versions.

5.3.3 Asynchronous Execution

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

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

5.3.4 Model Persistence

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

5.3.5 Examples

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

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

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

```python
from civis.ml import ModelPipeline
algorithms = ['gradient_boosting_classifier', 'sparse_logistic', 'random_forest_
→classifier']
pkey = 'person_id'
depvar = 'likes_cats'
```

(continues on next page)

```
models = [ModelPipeline(alg, primary_key=pkey, dependent_variable=depvar) for alg in
↪algorithms]
train = [model.train(table_name='schema.name', database_name='My DB') for model in
↪models]
aucs = [tr.metrics['roc_auc'] for tr in train]  # Code blocks here
```

5.3.6 Registering Models Trained Outside of Civis

Instead of using CivisML to train your model, you may train any scikit-learn-compatible model outside of Civis Platform and use `civis.ml.ModelPipeline.register_pretrained_model()` to register it as a CivisML model in Civis Platform. This will let you use Civis Platform to make predictions using your model, either to take advantage of distributed predictions on large datasets, or to create predictions as part of a workflow or service in Civis Platform.

When registering a model trained outside of Civis Platform, you are strongly advised to provide an ordered list of feature names used for training. This will allow CivisML to ensure that tables of data input for predictions have the correct features in the correct order. If your model has more than one output, you should also provide a list of output names so that CivisML knows how many outputs to expect and how to name them in the resulting table of model predictions.

If your model uses dependencies which aren't part of the default CivisML execution environment, you must provide them to the `dependencies` parameter of the `register_pretrained_model()` function, just as with the `ModelPipeline` constructor.

5.3.7 Sharing Models

Models produced by CivisML can't be shared directly through the Civis Platform UI or API. The `ml` namespace provides functions which will let you share your CivisML models with other Civis Platform users. To share your models, use the functions

- *put_models_shares_users()*
- *put_models_shares_groups()*
- *delete_models_shares_users()*
- *delete_models_shares_groups()*

To find out what models a user has, use *list_models()*.

5.3.8 Object and Function Reference

class civis.ml.**ModelPipeline**(*model*, *dependent_variable*, *primary_key=None*, *parameters=None*, *cross_validation_parameters=None*, *model_name=None*, *calibration=None*, *excluded_columns=None*, *client=None*, *cpu_requested=None*, *memory_requested=None*, *disk_requested=None*, *notifications=None*, *dependencies=None*, *git_token_name=None*, *verbose=False*, *etl=None*, *civisml_version=None*)

Interface for scikit-learn modeling in the Civis Platform

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

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

> **Parameters**

model [string or Estimator] Either the name of a pre-defined model (e.g. "sparse_logistic" or "gradient_boosting_classifier") or else a pre-existing Estimator object.

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

primary_key [string, optional] The unique ID (primary key) of the training dataset. This will be used to index the out-of-sample scores.

parameters [dict, optional] Specify parameters for the final stage estimator in a predefined model, e.g. `{'C': 2}` for a "sparse_logistic" model.

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

model_name [string, optional] The prefix of the Platform modeling jobs. It will have " Train" or " Predict" added to become the Script title.

calibration [{None, "sigmoid", "isotonic"}] If not None, calibrate output probabilities with the selected method. Valid only with classification models.

excluded_columns [array, optional] A list of columns which will be considered ineligible to be independent variables.

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

cpu_requested [int, optional] Number of CPU shares requested in the Civis Platform for training jobs. 1024 shares = 1 CPU.

memory_requested [int, optional] Memory requested from Civis Platform for training jobs, in MiB

disk_requested [float, optional] Disk space requested on Civis Platform for training jobs, in GB

notifications [dict] See *post_custom()* for further documentation about email and URL notification.

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

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

verbose [bool, optional] If True, supply debug outputs in Platform logs and make prediction child jobs visible.

etl [Estimator, optional] Custom ETL estimator which overrides the default ETL, and is run before training and validation.

civisml_version [str, optional] CivisML version to use for training and prediction. If not provided, the latest version in production is used.

See also:

civis.ml.ModelFuture

Examples

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

Attributes

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

Methods

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

classmethod from_existing(*train_job_id*, *train_run_id='latest'*, *client=None*)
> Create a *ModelPipeline* object from existing model IDs

> **Parameters**

> > **train_job_id** [int] The ID of the CivisML job in the Civis Platform

train_run_id [int or string, optional] Location of the model run, either

- an explicit run ID,

- "latest" : The most recent run

- "active" : The run designated by the training job's "active build" parameter

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

Returns

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

Examples

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

predict(*df=None*, *csv_path=None*, *table_name=None*, *database_name=None*, *manifest=None*, *file_id=None*, *sql_where=None*, *sql_limit=None*, *primary_key=Sentinel()*, *output_table=None*, *output_db=None*, *if_exists='fail'*, *n_jobs=None*, *polling_interval=None*, *cpu=None*, *memory=None*, *disk_space=None*, *dvs_to_predict=None*)
Make predictions on a trained model

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

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

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

Parameters

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

csv_path [str, optional] The location of a CSV of data on the local disk. It will be uploaded to a Civis file.

table_name [str, optional] The qualified name of the table containing your data

database_name [str, optional] Name of the database holding the data, e.g., 'My Redshift Cluster'.

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

file_id [int, optional] If the data are a CSV stored in a Civis file, provide the integer file ID.

sql_where [str, optional] A SQL WHERE clause used to scope the rows to be predicted

sql_limit [int, optional] SQL LIMIT clause to restrict the size of the prediction set

primary_key [str, optional] Primary key of the prediction table. Defaults to the primary key of the training data. Use `None` to indicate that the prediction data don't have a primary key column.

output_table: str, optional The table in which to put the predictions.

output_db [str, optional] Database of the output table. Defaults to the database of the input table.

if_exists [{'fail', 'append', 'drop', 'truncate'}] Action to take if the prediction table already exists.

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

polling_interval [float, optional] Check for job completion every this number of seconds. Do not set if using the notifications endpoint.

cpu [int, optional] CPU shares requested by the user for a single job.

memory [int, optional] RAM requested by the user for a single job.

disk_space [float, optional] disk space requested by the user for a single job.

dvs_to_predict [list of str, optional] If this is a multi-output model, you may list a subset of dependent variables for which you wish to generate predictions. This list must be a subset of the original *dependent_variable* input. The scores for the returned subset will be identical to the scores which those outputs would have had if all outputs were written, but ignoring some of the model's outputs will let predictions complete faster and use less disk space. The default is to produce scores for all DVs.

Returns

ModelFuture

classmethod register_pretrained_model(*model*, *dependent_variable=None*, *features=None*, *primary_key=None*, *model_name=None*, *dependencies=None*, *git_token_name=None*, *skip_model_check=False*, *verbose=False*, *client=None*, *civisml_version=None*)

Use a fitted scikit-learn model with CivisML scoring

Use this function to set up your own fitted scikit-learn-compatible Estimator object for scoring with CivisML. This function will upload your model to Civis Platform and store enough metadata about it that you can subsequently use it with a CivisML scoring job.

The only required input is the model itself, but you are strongly recommended to also provide a list of feature names. Without a list of feature names, CivisML will have to assume that your scoring table contains only the features needed for scoring (perhaps also with a primary key column), in all in the correct order.

Parameters

model [sklearn.base.BaseEstimator or int] The model object. This must be a fitted scikit-learn compatible Estimator object, or else the integer Civis File ID of a pickle or joblib-serialized file which stores such an object. If an Estimator object is provided, it will be uploaded to the Civis Files endpoint and set to be available indefinitely.

dependent_variable [string or List[str], optional] The dependent variable of the training dataset. For a multi-target problem, this should be a list of column names of dependent variables.

features [string or List[str], optional] A list of column names of features which were used for training. These will be used to ensure that tables input for prediction have the correct features in the correct order.

primary_key [string, optional] The unique ID (primary key) of the scoring dataset

model_name [string, optional] The name of the Platform registration job. It will have " Predict" added to become the Script title for predictions.

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

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

skip_model_check [bool, optional] If you're sure that your model will work with CivisML, but it will fail the comprehensive verification, set this to True.

verbose [bool, optional] If True, supply debug outputs in Platform logs and make prediction child jobs visible.

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

civisml_version [str, optional] CivisML version to use. If not provided, the latest version in production is used.

Returns

ModelPipeline

Examples

This example assumes that you already have training data `X` and `y`, where `X` is a `DataFrame`.

```
>>> from civis.ml import ModelPipeline
>>> from sklearn.linear_model import Lasso
>>> est = Lasso().fit(X, y)
>>> model = ModelPipeline.register_pretrained_model(
...     est, 'concrete', features=X.columns)
>>> model.predict(table_name='my.table', database_name='my-db')
```

train(*df=None, csv_path=None, table_name=None, database_name=None, file_id=None, sql_where=None, sql_limit=None, oos_scores=None, oos_scores_db=None, if_exists='fail', fit_params=None, polling_interval=None, validation_data='train', n_jobs=None*)

Start a Civis Platform job to train your model

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

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

Parameters

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

csv_path [str, optional] The location of a CSV of data on the local disk. It will be uploaded to a Civis file.

table_name [str, optional] The qualified name of the table containing the training set from which to build the model.

database_name [str, optional] Name of the database holding the training set table used to build the model. E.g., 'My Cluster Name'.

file_id [int, optional] If the training data are stored in a Civis file, provide the integer file ID.

sql_where [str, optional] A SQL WHERE clause used to scope the rows of the training set (used for table input only)

sql_limit [int, optional] SQL LIMIT clause for querying the training set (used for table input only)

oos_scores [str, optional] If provided, store out-of-sample predictions on training set data to this Redshift "schema.tablename".

oos_scores_db [str, optional] If not provided, store OOS predictions in the same database which holds the training data.

if_exists [{'fail', 'append', 'drop', 'truncate'}] Action to take if the out-of-sample prediction table already exists.

fit_params: Dict[str, str] Mapping from parameter names in the model's `fit` method to the column names which hold the data, e.g. `{'sample_weight': 'survey_weight_column'}`.

polling_interval [float, optional] Check for job completion every this number of seconds. Do not set if using the notifications endpoint.

validation_data [str, optional] Source for validation data. There are currently two options: *'train'* (the default), which cross-validates over training data for validation; and *'skip'*, which skips the validation step.

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

Returns

ModelFuture

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

    Encapsulates asynchronous execution of a CivisML job

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

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

    Parameters

        job_id [int] ID of the modeling job

        run_id [int] ID of the modeling run

        train_job_id [int, optional] If not provided, this object is assumed to encapsulate a training job, and `train_job_id` will equal `job_id`.

        train_run_id [int, optional] If not provided, this object is assumed to encapsulate a training run, and `train_run_id` will equal `run_id`.

        polling_interval [int or float, optional] The number of seconds between API requests to check whether a result is ready.

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

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

    See also:

civis.futures.CivisFuture

civis.futures.ContainerFuture

concurrent.futures.Future

    Attributes

        metadata [dict, blocking] The metadata associated with this modeling job

        metrics [dict, blocking] Validation metrics from this job's training

        validation_metadata [dict, blocking] Metadata from this modeling job's validation run

        train_metadata [dict, blocking] Metadata from this modeling job's training run (will be identical to *metadata* if this is a training run)

        estimator [sklearn.pipeline.Pipeline, blocking] The fitted scikit-learn Pipeline resulting from this model run

        table [pandas.DataFrame, blocking] The table output from this modeling job: out-of-sample predictions on the training set for a training job, or a table of predictions for a prediction job. If the prediction job was split into multiple files (this happens automatically for large tables), this attribute will provide only predictions for the first file.

        state [str] The current state of the Civis Platform run

        job_id [int]

        run_id [int]

train_job_id [int] Container ID for the training job – identical to `job_id` if this is a training job.

train_run_id [int] As `train_job_id` but for runs

is_training [bool] True if this `ModelFuture` corresponds to a train-validate job.

Methods

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

add_done_callback(*fn*)
> Attaches a callable that will be called when the future finishes.

> **Args:**

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

cancel()
> Submit a request to cancel the container/script/run.

>> **Returns**

>> **bool** Whether or not the job is in a cancelled state.

cancelled()
> Return True if the future was cancelled.

done()
> Return True of the future was cancelled or finished executing.

exception(*timeout=None*)
> Return the exception raised by the call that the future represents.

> **Args:**

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

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

> **Raises:** CancelledError: If the future was cancelled. TimeoutError: If the future didn't finish executing before the given

>> timeout.

failed()
> Return `True` if the Civis job failed.

`outputs()`
>   Block on job completion and return a list of run outputs.
>
>   The method will only return run outputs for successful jobs. Failed jobs will raise an exception.
>
>>   **Returns**
>>
>>>   **list[dict]** List of run outputs from a successfully completed job.
>>
>>   **Raises**
>>
>>>   **civis.base.CivisJobFailure** If the job fails.

`result`(*timeout=None*)
>   Return the result of the call that the future represents.
>
>   **Args:**
>
>>   **timeout: The number of seconds to wait for the result if the future** isn't done. If None, then there is no limit on the wait time.
>
>   **Returns:** The result of the call that the future represents.
>
>   **Raises:** CancelledError: If the future was cancelled. TimeoutError: If the future didn't finish executing before the given
>
>>   timeout.
>
>   Exception: If the call raised then that exception will be raised.

`running()`
>   Return True if the future is currently executing.

`set_exception`(*exception*)
>   Sets the result of the future as being the given exception.
>
>   This is adapted from https://github.com/python/cpython/blob/3.8/Lib/concurrent/futures/_base.py#L532-L545 This version does not try to change the _state or check that the initial _state is running since the Civis implementation has _state depend on the Platform job state.

`set_result`(*result*)
>   Sets the return value of work associated with the future.
>
>   This is adapted from https://github.com/python/cpython/blob/3.8/Lib/concurrent/futures/_base.py#L517-L530 This version does not try to change the _state or check that the initial _state is running since the Civis implementation has _state depend on the Platform job state.

`set_running_or_notify_cancel()`
>   Mark the future as running or process any cancel notifications.
>
>   Should only be used by Executor implementations and unit tests.
>
>   If the future has been cancelled (cancel() was called and returned True) then any threads waiting on the future completing (though calls to as_completed() or wait()) are notified and False is returned.
>
>   If the future was not cancelled then it is put in the running state (future calls to running() will return True) and True is returned.
>
>   This method should be called by Executor implementations before executing the work associated with this future. If this method returns False then the work should not be executed.
>
>   **Returns:** False if the Future was cancelled, True otherwise.
>
>   **Raises:**
>
>>   **RuntimeError: if this method was already called or if set_result**() or set_exception() was called.

> **succeeded**()
>> Return `True` if the job completed in Civis with no error.

civis.ml.**put_models_shares_users**(*id*, *user_ids*, *permission_level*, *client=None*,
                                    share_email_body='DEFAULT', *send_shared_email='DEFAULT'*)
> Set the permissions users have on this object

> Use this on both training and scoring jobs. If used on a training job, note that "read" permission is sufficient to score the model.

>> **Parameters**

>>> **id** [integer] The ID of the resource that is shared.

>>> **user_ids** [list] An array of one or more user IDs.

>>> **permission_level** [string] Options are: "read", "write", or "manage".

>>> **client** [*civis.APIClient*, optional] If not provided, an *civis.APIClient* object will be created from the CIVIS_API_KEY.

>>> **share_email_body** [string, optional] Custom body text for e-mail sent on a share.

>>> **send_shared_email** [boolean, optional] Send email to the recipients of a share.

>> **Returns**

>>> **readers** [dict::]

>>>> • **users** [list::]

>>>>> – id : integer

>>>>> – name : string

>>>> • **groups** [list::]

>>>>> – id : integer

>>>>> – name : string

>>> **writers** [dict::]

>>>> • **users** [list::]

>>>>> – id : integer

>>>>> – name : string

>>>> • **groups** [list::]

>>>>> – id : integer

>>>>> – name : string

>>> **owners** [dict::]

>>>> • **users** [list::]

>>>>> – id : integer

>>>>> – name : string

>>>> • **groups** [list::]

>>>>> – id : integer

>>>>> – name : string

total_user_shares [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

total_group_shares [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

civis.ml.**put_models_shares_groups**(*id*, *group_ids*, *permission_level*, *client=None*, *share_email_body='DEFAULT'*, *send_shared_email='DEFAULT'*)

Set the permissions groups have on this model.

Use this on both training and scoring jobs. If used on a training job, note that "read" permission is sufficient to score the model.

Parameters

id [integer] The ID of the resource that is shared.

group_ids [list] An array of one or more group IDs.

permission_level [string] Options are: "read", "write", or "manage".

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

share_email_body [string, optional] Custom body text for e-mail sent on a share.

send_shared_email [boolean, optional] Send email to the recipients of a share.

Returns

readers [dict::]

- **users** [list::]

  – id : integer

  – name : string

- **groups** [list::]

  – id : integer

  – name : string

writers [dict::]

- **users** [list::]

  – id : integer

  – name : string

- **groups** [list::]

  – id : integer

  – name : string

owners [dict::]

- **users** [list::]

  – id : integer

  – name : string

- **groups** [list::]

  – id : integer

>   **–** name : string

      total_user_shares [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

      total_group_shares [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

civis.ml.**delete_models_shares_users**(*id*, *user_id*, *client=None*)

    Revoke the permissions a user has on this object

    Use this function on both training and scoring jobs.

>   **Parameters**
>
>   **id** [integer] The ID of the resource that is shared.
>
>   **user_id** [integer] The ID of the user.
>
>   **client** [`civis.APIClient`, optional] If not provided, an `civis.APIClient` object will be created from the CIVIS_API_KEY.
>
>   **Returns**
>
>   **None** Response code 204: success

civis.ml.**delete_models_shares_groups**(*id*, *group_id*, *client=None*)

    Revoke the permissions a group has on this object

    Use this function on both training and scoring jobs.

>   **Parameters**
>
>   **id** [integer] The ID of the resource that is shared.
>
>   **group_id** [integer] The ID of the group.
>
>   **client** [`civis.APIClient`, optional] If not provided, an `civis.APIClient` object will be created from the CIVIS_API_KEY.
>
>   **Returns**
>
>   **None** Response code 204: success

civis.ml.**list_models**(*job_type='train'*, *author=Sentinel()*, *client=None*, ***kwargs*)

    List a user's CivisML models.

>   **Parameters**
>
>   **job_type** [{"train", "predict", None}] The type of model job to list. If "train", list training jobs only (including registered models trained outside of CivisML). If "predict", list prediction jobs only. If None, list both.
>
>   **author** [int, optional] User id of the user whose models you want to list. Defaults to the current user. Use None to list models from all users.
>
>   **client** [`civis.APIClient`, optional] If not provided, an `civis.APIClient` object will be created from the CIVIS_API_KEY.
>
>   ****kwargs** [kwargs] Extra keyword arguments passed to *client.scripts.list_custom()*

    See also:

    `APIClient.scripts.list_custom`

5.4 Parallel Computation

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

5.4.1 Joblib backend

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

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

- Each function call which is parallelized with the Civis joblib backend will run in a different Civis Platform script. Creating a new script comes with some overhead. It will take between a few seconds and a few minutes for each script to start, depending on whether Civis Platform needs to provision additional resources. If you expect that each function call will complete quickly, instead consider either running them in serial or using extra processes in the same Civis Platform script.

- Because function calls run in different scripts, function inputs and outputs must be uploaded to Civis Platform from their origin script and downloaded into their destination. If your functions take very large inputs and/or produce very large outputs, moving the data around will cause additional overhead. Consider either using a different tool or refactoring your code so that the function to be parallelized is no longer moving around large amounts of data.

- Some open-source libraries, such as `scikit-learn`, use `joblib` to do computations in parallel. If you're working with such a library, the Civis joblib backend provides an easy way to run these parallel computations in different Civis Platform scripts.

Joblib

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

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

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

How to use

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

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

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

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

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

You can find more about custom joblib backends in the joblib documentation.

Note that `joblib.Parallel` takes both a `n_jobs` and `pre_dispatch` parameter. The Civis joblib backend doesn't queue submitted jobs itself, so it will run `pre_dispatch` jobs at once.

Note: The default value of `pre_dispatch` is `"2*n_jobs"`, which will run a maximum of `2 * n_jobs` jobs at once on Civis Platform. Set `pre_dispatch="n_jobs"` in your `joblib.Parallel` call to run at most `n_jobs` jobs.

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

The joblib backend will automatically add environment variables called "CIVIS_PARENT_JOB_ID" and "CIVIS_PARENT_RUN_ID", holding the values of the job and run IDs of the Civis Platform job in which you're running the joblib backend (if any). Your functions could use these to communicate with the parent job or to recognize that they're in a process which has been created by another Civis Platform job. However, where possible you should let the joblib backend itself transport the return value of the function it's running back to the parent.

Infer backend parameters

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

Templated Scripts

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

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

Examples

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

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

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

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

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

```
>>> from joblib import parallel_backend, register_parallel_backend
>>> from sklearn.model_selection import GridSearchCV
>>> from sklearn.ensemble import GradientBoostingClassifier
>>> from sklearn.datasets import load_digits
>>> digits = load_digits()
>>> param_grid = {
...     "max_depth": [1, 3, 5, None],
...     "max_features": ["sqrt", "log2", None],
...     "learning_rate": [0.1, 0.01, 0.001]
... }
>>> # Note: n_jobs and pre_dispatch specify the maximum number of
```

(continues on next page)

```
>>> # concurrent jobs.
>>> gs = GridSearchCV(GradientBoostingClassifier(n_estimators=1000,
...                                               random_state=42),
...                   param_grid=param_grid,
...                   n_jobs=5, pre_dispatch="n_jobs")
>>> register_parallel_backend('civis', make_backend_factory(
...     required_resources={"cpu": 512, "memory": 256}))
>>> with parallel_backend('civis'):
...     gs.fit(digits.data, digits.target)
```

Debugging

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

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

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

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

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

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

5.4.2 Object Reference

Parallel computations using the Civis Platform infrastructure

exception civis.parallel.**JobSubmissionError**

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

Infer the container environment and return a backend factory.

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

Jobs created through this backend will have environment variables "CIVIS_PARENT_JOB_ID" and "CIVIS_PARENT_RUN_ID" with the contents of the "CIVIS_JOB_ID" and "CIVIS_RUN_ID" of the environment which created them. If the code doesn't have "CIVIS_JOB_ID" and "CIVIS_RUN_ID" environment

variables available, the child will not have "CIVIS_PARENT_JOB_ID" and "CIVIS_PARENT_RUN_ID" environment variables.

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

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

> **Parameters**
>
> > **required_resources** [dict or None, optional] The resources needed by the container. See the *container scripts API documentation <https://platform.civisanalytics.com/api#resources-scripts>* for details. Resource requirements not specified will default to the requirements of the current job.
> >
> > **params** [list or None, optional] A definition of the parameters this script accepts in the arguments field. See the *container scripts API documentation <https://platform.civisanalytics.com/api#resources-scripts>* for details.
> >
> > Parameters of the child jobs will default to the parameters of the current job. Any parameters provided here will override parameters of the same name from the current job.
> >
> > **arguments** [dict or None, optional] Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params. See the *container scripts API documentation <https://platform.civisanalytics.com/api#resources-scripts>* for details.
> >
> > Arguments will default to the arguments of the current job. Anything provided here will override portions of the current job's arguments.
> >
> > **client** [*civis.APIClient* instance or None, optional] An API Client object to use.
> >
> > **polling_interval** [int, optional] The polling interval, in seconds, for checking container script status. If you have many jobs, you may want to set this higher (e.g., 300) to avoid *rate-limiting <https://platform.civisanalytics.com/api#basics>*.
> >
> > **setup_cmd** [str, optional] A shell command or sequence of commands for setting up the environment. These will precede the commands used to run functions in joblib. This is primarily for installing dependencies that are not available in the dockerhub repo (e.g., "cd /app && python setup.py install" or "pip install gensim").
> >
> > With no GitHub repo input, the setup command will default to a command that does nothing. If a `repo_http_uri` is provided, the default setup command will attempt to run "python setup.py install". If this command fails, execution will still continue.
> >
> > **max_submit_retries** [int, optional] The maximum number of retries for submitting each job. This is to help avoid a large set of jobs failing because of a single 5xx error. A value higher than zero should only be used for jobs that are idempotent (i.e., jobs whose result and side effects are the same regardless of whether they are run once or many times).
> >
> > **max_job_retries** [int, optional] Retry failed jobs this number of times before giving up. Even more than with `max_submit_retries`, this should only be used for jobs which are idempotent, as the job may have caused side effects (if any) before failing. These retries assist with jobs which may have failed because of network or worker failures.
> >
> > **hidden: bool, optional** The hidden status of the object. Setting this to true hides it from most API endpoints. The object can still be queried directly by ID. Defaults to True.

> **remote_backend** [str or object, optional] The name of a joblib backend or a joblib backend
> itself. This parameter is the joblib backend to use when executing code within joblib
> in the container. The default of 'sequential' uses the joblib sequential backend in the
> container. The value 'civis' uses an exact copy of the Civis joblib backend that launched
> the container. Note that with the value 'civis', one can potentially use more jobs than
> specified by `n_jobs`.
>
> ****kwargs:** Additional keyword arguments will be passed directly to `post_containers()`,
> potentially overriding the values of those arguments in the parent environment.

> **Raises**

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

See also:

civis.parallel.make_backend_factory

civis.parallel.**make_backend_factory**(*docker_image_name='civisanalytics/datascience-python'*,
  client=None, *polling_interval=None*, *setup_cmd=None*,
  max_submit_retries=0, *max_job_retries=0*, *hidden=True*,
  remote_backend='sequential', ***kwargs*)
Create a joblib backend factory that uses Civis Container Scripts

Jobs created through this backend will have environment variables "CIVIS_PARENT_JOB_ID" and
"CIVIS_PARENT_RUN_ID" with the contents of the "CIVIS_JOB_ID" and "CIVIS_RUN_ID" of the environment which created them. If the code doesn't have "CIVIS_JOB_ID" and "CIVIS_RUN_ID" environment
variables available, the child will not have "CIVIS_PARENT_JOB_ID" and "CIVIS_PARENT_RUN_ID" environment variables.

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

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

> **Parameters**

> **docker_image_name** [str, optional] The image for the container script. You may also wish to
> specify a `docker_image_tag` in the keyword arguments.

> **client** [*civis.APIClient* instance or None, optional] An API Client object to use.

> **polling_interval** [int, optional] The polling interval, in seconds, for checking container script
> status. If you have many jobs, you may want to set this higher (e.g., 300) to avoid *rate-
> limiting <https://platform.civisanalytics.com/api#basics>*.

> **setup_cmd** [str, optional] A shell command or sequence of commands for setting up the en-
> vironment. These will precede the commands used to run functions in joblib. This is
> primarily for installing dependencies that are not available in the dockerhub repo (e.g.,
> "cd /app && python setup.py install" or "pip install gensim").
>
> With no GitHub repo input, the setup command will default to a command that does
> nothing. If a *repo_http_uri* is provided, the default setup command will attempt to run
> "python setup.py install". If this command fails, execution will still continue.

---

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

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

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

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

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

See also:

`civis.APIClient.scripts.post_containers`

Notes

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

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

Examples

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

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

```
>>> # Using the Civis backend:
>>> from joblib import parallel_backend, register_parallel_backend
>>> from civis.parallel import make_backend_factory
>>> register_parallel_backend('civis', make_backend_factory(
...     required_resources={"cpu": 512, "memory": 256}))
>>> with parallel_backend('civis'):
...     parallel = Parallel(n_jobs=5, pre_dispatch='n_jobs')
...     print(parallel(delayed(sqrt)(i ** 2) for i in range(10)))
[0.0, 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0]
```

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

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

Create a joblib backend factory that uses Civis Custom Scripts.

If your template has settable parameters "CIVIS_PARENT_JOB_ID" and "CIVIS_PARENT_RUN_ID", then this executor will fill them with the contents of the "CIVIS_JOB_ID" and "CIVIS_RUN_ID" of the environment which created them. If the code doesn't have "CIVIS_JOB_ID" and "CIVIS_RUN_ID" environment variables available, the child will not have "CIVIS_PARENT_JOB_ID" and "CIVIS_PARENT_RUN_ID" environment variables.

Parameters

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

arguments [dict or None, optional] Dictionary of name/value pairs to use to run this script. Only settable if this script has defined params. See the *container scripts API documentation <https://platform.civisanalytics.com/api#resources-scripts>* for details.

client [*civis.APIClient* instance or None, optional] An API Client object to use.

polling_interval [int, optional] The polling interval, in seconds, for checking container script status. If you have many jobs, you may want to set this higher (e.g., 300) to avoid *rate-limiting <https://platform.civisanalytics.com/api#basics>*.

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

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

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

5.5 API Client

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

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

The methods on *APIClient* are created dynamically at runtime by parsing an `collections.OrderedDict` representation of the Civis API specification. The methods are generated based on the path and HTTP method used with each endpoint. For example, `GET /workflows/1` can be accessed with `client.workflows.get(1)`. GET endpoints that don't end in a parameter use a `list` method instead. Below are examples of endpoints and how they map to API Client methods:

Endpoint	API Client Method
GET /workflows	client.workflows.list()
GET /workflows/1	client.workflows.get(1)
GET /workflows/1/executions	client.workflows.list_executions(1)
PATCH /workflows/1	client.workflows.patch(1, ...)
POST /workflows/1/executions	client.workflows.post_executions(1)
GET /workflows/1/executions/2	client.workflows.get_executions(1, 2)

Note that Python's built-in `help` function can be used to see lists of available endpoints for a resource (e.g., `help(client.workflows)`) or to get documentation for a specific endpoint function (e.g., `help(client.workflows.list)`). The ? operator in IPython (e.g., `?client.workflows`) and the `shift-tab` hotkey in a Jupyter notebook also cause documentation to be displayed.

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

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

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

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

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

    The Civis API client.

        Parameters

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

            return_type [str, optional] The following types are implemented:

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

- `'snake'` Returns a `civis.response.Response` object for the json-encoded content of a response. This maps the top-level json keys to snake_case.

- `'pandas'` Returns a `pandas.DataFrame` for list-like responses and a `pandas.Series` for single a json response.

            retry_total [DEPRECATED int, optional] A number indicating the maximum number of retries for 429, 502, 503, or 504 errors. This parameter no longer has any effect since v1.15.0, as retries are automatically handled. This parameter will be removed at version 2.0.0.

            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. This will be removed in a future version of the API client.

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

        Attributes

            admin An instance of the *Admin* endpoint

            aliases An instance of the *Aliases* endpoint

            announcements An instance of the *Announcements* endpoint

            clusters An instance of the *Clusters* endpoint

            credentials An instance of the *Credentials* endpoint

databases An instance of the [*Databases*](#) endpoint

endpoints An instance of the [*Endpoints*](#) endpoint

enhancements An instance of the [*Enhancements*](#) endpoint

exports An instance of the [*Exports*](#) endpoint

files An instance of the [*Files*](#) endpoint

git_repos An instance of the [*Git_Repos*](#) endpoint

groups An instance of the [*Groups*](#) endpoint

imports An instance of the [*Imports*](#) endpoint

jobs An instance of the [*Jobs*](#) endpoint

json_values An instance of the [*Json_Values*](#) endpoint

match_targets An instance of the [*Match_Targets*](#) endpoint

media An instance of the [*Media*](#) endpoint

models An instance of the [*Models*](#) endpoint

notebooks An instance of the [*Notebooks*](#) endpoint

notifications An instance of the [*Notifications*](#) endpoint

ontology An instance of the [*Ontology*](#) endpoint

permission_sets An instance of the [*Permission_Sets*](#) endpoint

predictions An instance of the [*Predictions*](#) endpoint

projects An instance of the [*Projects*](#) endpoint

queries An instance of the [*Queries*](#) endpoint

remote_hosts An instance of the [*Remote_Hosts*](#) endpoint

reports An instance of the [*Reports*](#) endpoint

roles An instance of the [*Roles*](#) endpoint

saml_service_providers An instance of the [*Saml_Service_Providers*](#) endpoint

scripts An instance of the [*Scripts*](#) endpoint

search An instance of the [*Search*](#) endpoint

services An instance of the [*Services*](#) endpoint

storage_hosts An instance of the [*Storage_Hosts*](#) endpoint

table_tags An instance of the [*Table_Tags*](#) endpoint

tables An instance of the [*Tables*](#) endpoint

templates An instance of the [*Templates*](#) endpoint

users An instance of the [*Users*](#) endpoint

workflows An instance of the [*Workflows*](#) endpoint

Methods

get_aws_credential_id(cred_name[, owner])	Find an AWS credential ID.
get_database_credential_id(username, ...)	Return the credential ID for a given username in a given database.
get_database_id(database)	Return the database ID for a given database name.
get_storage_host_id(storage_host)	Return the storage host ID for a given storage host name.
get_table_id(table, database)	Return the table ID for a given database and table name.

property 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_storage_host_id(*storage_host*)
> Return the storage host ID for a given storage host name.

> **Parameters**

>> **storage_host** [str or int] If an integer ID is given, passes through. If a str is given the storage host ID corresponding to that storage host is returned.

> **Returns**

>> **storage_host_id** [int] The ID of the storage host.

> **Raises**

>> **ValueError** If the storage host can't be found.

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.get_storage_host_id('test host')
1234
```

```
>>> client.get_storage_host_id(1111)
1111
```

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.tablename. Either schema or table-name, or both, can be double-quoted to correctly parse special characters (such as '.').
> >
> > **database** [str or int] The name or ID of the database.
>
> **Returns**
>
> > **table_id** [int] The ID of the table.
>
> **Raises**
>
> > **ValueError** If a table match can't be found.

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.get_table_id('foo.bar', 'redshift-general')
123
>>> client.get_table_id('"schema.has.periods".bar', 'redshift-general')
456
```

property username

The current user's username.

5.5.1 API Responses

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 | None] This is *json_data* as it is originally returned to the user without the key names being changed. See Notes. None is used if the original response returned a 204 No Content response.
> >
> > **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.

Methods

clear()	
copy()	
fromkeys(iterable[, value])	Create a new dictionary with keys from iterable and values set to value.
get(key[, default])	Return the value for key if key is in the dictionary, else default.
items()	
keys()	
pop(k[,d])	If key is not found, d is returned if given, otherwise KeyError is raised
popitem()	2-tuple; but raise KeyError if D is empty.
setdefault(key[, default])	Insert key with a value of default if key is not in the dictionary.
update([E,]**F)	If E is present and has a .keys() method, then does: for k in E: D[k] = E[k] If E is present and lacks a .keys() method, then does: for k, v in E: D[k] = v In either case, this is followed by: for k in F: D[k] = F[k]
values()	

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

> A response object which is an iterator
>
> > **Parameters**
> >
> > > **path** [str] Make GET requests to this path.
> > >
> > > **initial_params** [dict] Query params that should be passed along with each request. Note that if *initial_params* contains the keys *page_num* or *limit*, they will be ignored. The given dict is not modified.
> > >
> > > **endpoint** [*civis.base.Endpoint*] An endpoint used to make API requests.

Notes

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

Examples

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

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

A class for tracking future results.

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

Parameters

poller [func] A function which returns an object that has a `state` attribute.

poller_args [tuple] The arguments with which to call the poller function.

polling_interval [int or float, optional] The number of seconds between API requests to check whether a result is ready.

api_key [DEPRECATED str, optional] Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.

client [`civis.APIClient`, optional]

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

Examples

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

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

```
>>> future.run_id == response.last_run_id
True
```

Attributes

job_id [int] First element of the tuple given to *poller_args*

run_id [int or None] Second element of the tuple given to *poller_args* (*None* if the poller function does not require a run ID)

Methods

add_done_callback(fn)	Attaches a callable that will be called when the future finishes.
cancel()	Not currently implemented.
cancelled()	Return True if the future was cancelled.
done()	Return True of the future was cancelled or finished executing.
exception([timeout])	Return the exception raised by the call that the future represents.
failed()	Return `True` if the Civis job failed.
outputs()	Block on job completion and return a list of run outputs.
result([timeout])	Return the result of the call that the future represents.
running()	Return True if the future is currently executing.
set_exception(exception)	Sets the result of the future as being the given exception.
set_result(result)	Sets the return value of work associated with the future.
set_running_or_notify_cancel()	Mark the future as running or process any cancel notifications.
succeeded()	Return `True` if the job completed in Civis with no error.

cleanup	

outputs()

Block on job completion and return a list of run outputs.

The method will only return run outputs for successful jobs. Failed jobs will raise an exception.

Returns

list[dict] List of run outputs from a successfully completed job.

Raises

civis.base.CivisJobFailure If the job fails.

Helper Functions

civis.**find**(*object_list*, *filter_func=None*, ***kwargs*)

> Filter *civis.response.Response* objects.
>
> > **Parameters**
> >
> > > **object_list** [iterable] An iterable of arbitrary objects, particularly those with attributes that can be targeted by the filters in *kwargs*. A major use case is an iterable of *civis.response. Response* objects.
> > >
> > > **filter_func** [callable, optional] A one-argument function. If specified, *kwargs* are ignored. An *object* from the input iterable is kept in the returned list if and only if `bool(filter_func(object))` is `True`.
> > >
> > > ****kwargs** Key-value pairs for more fine-grained filtering; they cannot be used in conjunction with *filter_func*. All keys must be strings. For an *object* from the input iterable to be included in the returned list, all the *key`s must be attributes of `object*, plus any one of the following conditions for a given *key*:
> > >
> > > > • *value* is a one-argument function and `bool(value(getattr(object, key)))` is `True`
> > > >
> > > > • *value* is `True`
> > > >
> > > > • `getattr(object, key)` is equal to `value`
> >
> > **Returns**
> >
> > > list
>
> See also:
>
> *civis.find_one*
>
> ## Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> # creds is a list of civis.response.Response objects
>>> creds = client.credentials.list()
>>> # target_creds contains civis.response.Response objects
>>> # with the attribute 'name' == 'username'
>>> target_creds = find(creds, name='username')
```

civis.**find_one**(*object_list*, *filter_func=None*, ***kwargs*)

> Return one satisfying *civis.response.Response* object.
>
> The arguments are the same as those for *civis.find()*. If more than one object satisfies the filtering criteria, the first one is returned. If no satisfying objects are found, `None` is returned.
>
> > **Returns**
> >
> > > object or None
>
> See also:
>
> *civis.find*

5.5.2 API Resources

Admin

class Admin(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.admin.list_announcements(...)
```

Methods

delete_announcements(id)	Delete an announcement
get_announcements(id)	Get a particular announcement
list_announcements(*[, limit, page_num, ...])	List announcements
list_organizations(*[, status, org_type])	List organizations
patch_announcements(id, *[, subject, body, ...])	Edit an announcement
patch_themes(id, *[, name, ...])	Edit a theme
post_announcements(subject, body, *[, ...])	Post an announcement
post_themes(name, settings_json, *[, ...])	Create a theme

——

delete_announcements(*id*)
> Delete an announcement

> > **Parameters**

> > > **id** [integer] The ID of this announcement

> > **Returns**

> > > **None** Response code 204: success

get_announcements(*id*)
> Get a particular announcement

> > **Parameters**

> > > **id** [integer] The ID of this announcement

> > **Returns**

> > > *civis.response.Response*

> > > > • **id** [integer] The ID of this announcement

> > > > • **subject** [string] The subject of this announcement.

> > > > • **body** [string] The body of this announcement.

> > > > • **released_at** [string/date-time] The date and time this announcement was released.

> > > > • created_at : string/date-time

- updated_at : string/date-time

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

List announcements

Parameters

limit [integer, optional] Number of results to return. Defaults to 10. Maximum allowed
is 50.

page_num [integer, optional] Page number of the results to return. Defaults to the first
page, 1.

order [string, optional] The field on which to order the result set. Defaults to released_at.
Must be one of: released_at.

order_dir [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

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

Returns

civis.response.PaginatedResponse

- **id** [integer] The ID of this announcement

- **subject** [string] The subject of this announcement.

- **body** [string] The body of this announcement.

- **released_at** [string/date-time] The date and time this announcement was released.

- created_at : string/date-time

- updated_at : string/date-time

list_organizations(*, *status='DEFAULT'*, *org_type='DEFAULT'*)

List organizations

Parameters

status [array, optional] The status of the organization (active/trial/inactive).

org_type [array, optional] The organization type (platform/ads/survey_vendor/other).

Returns

civis.response.Response

- **id** [integer] The ID of this organization.

- **name** [string] The name of this organization.

- **slug** [string] The slug of this organization.

- **account_manager_id** [integer] The user ID of the Account Manager.

- **cs_specialist_id** [integer] The user ID of the Client Success Specialist.

- **status** [string] The status of the organization (active/trial/inactive).

- **org_type** [string] The organization type (platform/ads/survey_vendor/other).

- **custom_branding** [string] The custom branding settings.

- **contract_size** [integer] The monthly contract size.

- **max_analyst_users** [integer] The max number of full platform users for the org.

- **max_report_users** [integer] The max number of report-only platform users for the org.

- **vertical** [string] The business vertical that the organization belongs to.

- **cs_metadata** [string] Additional metadata about the organization in JSON format.

- **remove_footer_in_emails** [boolean] If true, emails sent by platform will not include Civis text.

- **salesforce_account_id** [string] The SalesForce Account ID for this organization.

- **tableau_site_id** [string] The Tableau Site ID for this organization.

- **fedramp_enabled** [boolean] Flag denoting whether this organization is FedRAMP compliant.

- **created_by_id** [integer] The ID of the user who created this organization

- **last_updated_by_id** [integer] The ID of the user who last updated this organization

- **advanced_settings** [dict::]

    - **dedicated_dj_pool_enabled** [boolean] If true, the Organization has a dedicated delayed jobs pool. Defaults to false.

- **tableau_refresh_history** [list] The number of tableau refreshes used this month.

patch_announcements(*id*, ***, *subject='DEFAULT'*, *body='DEFAULT'*, *released_at='DEFAULT'*)

> Edit an announcement
>> **Parameters**
>>
>>> **id** [integer] The ID of this announcement
>>>
>>> **subject** [string, optional] The subject of this announcement.
>>>
>>> **body** [string, optional] The body of this announcement.
>>>
>>> **released_at** [string/date-time, optional] The date and time this announcement was released.
>>
>> **Returns**
>>
>>> *civis.response.Response*
>>>
>>> - **id** [integer] The ID of this announcement
>>>
>>> - **subject** [string] The subject of this announcement.
>>>
>>> - **body** [string] The body of this announcement.
>>>
>>> - **released_at** [string/date-time] The date and time this announcement was released.
>>>
>>> - created_at : string/date-time
>>>
>>> - updated_at : string/date-time

patch_themes(*id*, ***, *name='DEFAULT'*, *organization_ids='DEFAULT'*, *settings_json='DEFAULT'*, *logo_file_id='DEFAULT'*)

> Edit a theme
>> **Parameters**
>>
>>> **id** [integer] The ID of this theme.

> **name** [string, optional] The name of this theme.
>
> **organization_ids** [list, optional] List of organization ID's allowed to see this theme.
>
> **settings_json** [string, optional] The JSON-encoded theme configuration.
>
> **logo_file_id** [string, optional] The ID of the logo image file.

> **Returns**
>
> > [*civis.response.Response*](#)
> >
> > - **id** [integer] The ID of this theme.
> >
> > - **name** [string] The name of this theme.
> >
> > - **organization_ids** [list] List of organization ID's allowed to use this theme.
> >
> > - **settings** [string] The theme configuration object.
> >
> > - **logo_file** [dict::]
> >
> >   - **id** [integer] The ID of the logo image file.
> >
> >   - **download_url** [string] The URL of the logo image file.
> >
> > - created_at : string/date-time
> >
> > - updated_at : string/date-time

post_announcements(*subject*, *body*, ***, *released_at='DEFAULT'*)

> Post an announcement
>
> > **Parameters**
> >
> > > **subject** [string] The subject of this announcement.
> > >
> > > **body** [string] The body of this announcement.
> > >
> > > **released_at** [string/date-time, optional] The date and time this announcement was released.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > - **id** [integer] The ID of this announcement
> > >
> > > - **subject** [string] The subject of this announcement.
> > >
> > > - **body** [string] The body of this announcement.
> > >
> > > - **released_at** [string/date-time] The date and time this announcement was released.
> > >
> > > - created_at : string/date-time
> > >
> > > - updated_at : string/date-time

post_themes(*name*, *settings_json*, ***, *organization_ids='DEFAULT'*, *logo_file_id='DEFAULT'*)

> Create a theme
>
> > **Parameters**
> >
> > > **name** [string] The name of this theme.
> > >
> > > **settings_json** [string] The JSON-encoded theme configuration.
> > >
> > > **organization_ids** [list, optional] List of organization ID's allowed to see this theme.
> > >
> > > **logo_file_id** [string, optional] The ID of the logo image file.

> **Returns**
>
> > [*civis.response.Response*](#)
> >
> > > - **id** [integer] The ID of this theme.
> > >
> > > - **name** [string] The name of this theme.
> > >
> > > - **organization_ids** [list] List of organization ID's allowed to use this theme.
> > >
> > > - **settings** [string] The theme configuration object.
> > >
> > > - **logo_file** [dict::]
> > >
> > >   > - **id** [integer] The ID of the logo image file.
> > >   >
> > >   > - **download_url** [string] The URL of the logo image file.
> > >
> > > - created_at : string/date-time
> > >
> > > - updated_at : string/date-time

Aliases

class **Aliases**(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.aliases.list_shares(...)
```

Methods

[*delete*](#)(id)	Delete an alias
[*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 an Alias
[*get_object_type*](#)(object_type, alias)	Get details about an alias within an FCO type
[*list*](#)(*[, object_type, limit, page_num, ...])	List Aliases
[*list_dependencies*](#)(id, *[, user_id])	List dependent objects for this object
[*list_shares*](#)(id)	List users and groups permissioned on this object
[*patch*](#)(id, *[, object_id, object_type, ...])	Update some attributes of this Alias
[*post*](#)(object_id, object_type, alias, *[, ...])	Create an Alias
[*put*](#)(id, object_id, object_type, alias, *[, ...])	Replace all attributes of this Alias
[*put_shares_groups*](#)(id, group_ids, ...[, ...])	Set the permissions groups has on this object
[*put_shares_users*](#)(id, user_ids, ...[, ...])	Set the permissions users have on this object
[*put_transfer*](#)(id, user_id, ...[, email_body, ...])	Transfer ownership of this object to another user

---

> **delete**(*id*)
>
> > Delete an alias
> >
> > > **Parameters**

> **id** [integer] The id of the Alias object.

> **Returns**

>> **None** Response code 204: success

delete_shares_groups(*id*, *group_id*)

> Revoke the permissions a group has on this object

>> **Parameters**

>>> **id** [integer] The ID of the resource that is shared.

>>> **group_id** [integer] The 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] The ID of the resource that is shared.

>>> **user_id** [integer] The ID of the user.

>> **Returns**

>>> **None** Response code 204: success

get(*id*)

> Get an Alias

>> **Parameters**

>>> **id** [integer]

>> **Returns**

>>> [*civis.response.Response*](#)

>>>> • **id** [integer] The id of the Alias object.

>>>> • **object_id** [integer] The id of the object

>>>> • **object_type** [string] The type of the object. Valid types include: cass_ncoa, container_script, geocode, python_script, r_script, salesforce_export, javascript_script, sql_script, project, notebook, workflow, template_script, template_report, service, report, tableau and service_report.

>>>> • **alias** [string] The alias of the object

>>>> • **user_id** [integer] The id of the user who created the alias

>>>> • **display_name** [string] The display name of the Alias object. Defaults to object name if not provided.

get_object_type(*object_type*, *alias*)

> Get details about an alias within an FCO type

>> **Parameters**

>>> **object_type** [string] The type of the object. Valid types include: cass_ncoa, container_script, geocode, python_script, r_script, salesforce_export, javascript_script, sql_script, project, notebook, workflow, template_script, template_report, service, report, tableau and service_report.

> **alias** [string] The alias of the object

Returns

[*civis.response.Response*](#)

- **id** [integer] The id of the Alias object.

- **object_id** [integer] The id of the object

- **object_type** [string] The type of the object. Valid types include: cass_ncoa, container_script, geocode, python_script, r_script, salesforce_export, javascript_script, sql_script, project, notebook, workflow, template_script, template_report, service, report, tableau and service_report.

- **alias** [string] The alias of the object

- **user_id** [integer] The id of the user who created the alias

- **display_name** [string] The display name of the Alias object. Defaults to object name if not provided.

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

Parameters

> **object_type** [string, optional] Filter results by object type. Pass multiple object types with a comma- separatedlist. Valid types include: cass_ncoa, container_script, geocode, python_script, r_script, salesforce_export, javascript_script, sql_script, project, notebook, workflow, template_script, template_report, service, report, tableau and service_report.
>
> **limit** [integer, optional] Number of results to return. Defaults to 50. Maximum allowed is 1000.
>
> **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
>
> **order** [string, optional] The field on which to order the result set. Defaults to id. Must be one of: id, object_type.
>
> **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

[*civis.response.PaginatedResponse*](#)

- **id** [integer] The id of the Alias object.

- **object_id** [integer] The id of the object

- **object_type** [string] The type of the object. Valid types include: cass_ncoa, container_script, geocode, python_script, r_script, salesforce_export, javascript_script, sql_script, project, notebook, workflow, template_script, template_report, service, report, tableau and service_report.

- **alias** [string] The alias of the object

- **user_id** [integer] The id of the user who created the alias

- **display_name** [string] The display name of the Alias object. Defaults to object name if not provided.

list_dependencies(*id*, ***, *user_id='DEFAULT'*)

List dependent objects for this object

Parameters

id [integer] The ID of the resource that is shared.

user_id [integer, optional] ID of target user

Returns

[*civis.response.Response*]

- **object_type** [string] Dependent object type

- **fco_type** [string] Human readable dependent object type

- **id** [integer] Dependent object ID

- **name** [string] Dependent object name, or nil if the requesting user cannot read this object

- **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user

- **shareable** [boolean] Whether or not the requesting user can share this object.

list_shares(*id*)

List users and groups permissioned on this object

Parameters

id [integer] The ID of the resource that is shared.

Returns

[*civis.response.Response*]

- **readers** [dict::]

  – **users** [list::]

     ∗ id : integer

     ∗ name : string

  – **groups** [list::]

     ∗ id : integer

     ∗ name : string

- **writers** [dict::]

  – **users** [list::]

     ∗ id : integer

     ∗ name : string

  – **groups** [list::]

     ∗ id : integer

* name : string

- **owners** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch(*id*, *, *object_id='DEFAULT'*, *object_type='DEFAULT'*, *alias='DEFAULT'*, *display_name='DEFAULT'*)

Update some attributes of this Alias

Parameters

id [integer] The id of the Alias object.

object_id [integer, optional] The id of the object

object_type [string, optional] The type of the object. Valid types include: cass_ncoa, container_script, geocode, python_script, r_script, salesforce_export, javascript_script, sql_script, project, notebook, workflow, template_script, template_report, service, report, tableau and service_report.

alias [string, optional] The alias of the object

display_name [string, optional] The display name of the Alias object. Defaults to object name if not provided.

Returns

[*civis.response.Response*](#)

- **id** [integer] The id of the Alias object.

- **object_id** [integer] The id of the object

- **object_type** [string] The type of the object. Valid types include: cass_ncoa, container_script, geocode, python_script, r_script, salesforce_export, javascript_script, sql_script, project, notebook, workflow, template_script, template_report, service, report, tableau and service_report.

- **alias** [string] The alias of the object

- **user_id** [integer] The id of the user who created the alias

- **display_name** [string] The display name of the Alias object. Defaults to object name if not provided.

post(*object_id*, *object_type*, *alias*, *, *display_name='DEFAULT'*)

Create an Alias

Parameters

object_id [integer] The id of the object

object_type [string] The type of the object. Valid types include: cass_ncoa, container_script, geocode, python_script, r_script, salesforce_export, javascript_script, sql_script, project, notebook, workflow, template_script, template_report, service, report, tableau and service_report.

alias [string] The alias of the object

display_name [string, optional] The display name of the Alias object. Defaults to object name if not provided.

> **Returns**

> > *civis.response.Response*

> > > - **id** [integer] The id of the Alias object.

> > > - **object_id** [integer] The id of the object

> > > - **object_type** [string] The type of the object. Valid types include: cass_ncoa, container_script, geocode, python_script, r_script, salesforce_export, javascript_script, sql_script, project, notebook, workflow, template_script, template_report, service, report, tableau and service_report.

> > > - **alias** [string] The alias of the object

> > > - **user_id** [integer] The id of the user who created the alias

> > > - **display_name** [string] The display name of the Alias object. Defaults to object name if not provided.

put(*id*, *object_id*, *object_type*, *alias*, ***, *display_name='DEFAULT'*)

> Replace all attributes of this Alias

> > **Parameters**

> > > **id** [integer] The id of the Alias object.

> > > **object_id** [integer] The id of the object

> > > **object_type** [string] The type of the object. Valid types include: cass_ncoa, container_script, geocode, python_script, r_script, salesforce_export, javascript_script, sql_script, project, notebook, workflow, template_script, template_report, service, report, tableau and service_report.

> > > **alias** [string] The alias of the object

> > > **display_name** [string, optional] The display name of the Alias object. Defaults to object name if not provided.

> > **Returns**

> > > *civis.response.Response*

> > > > - **id** [integer] The id of the Alias object.

> > > > - **object_id** [integer] The id of the object

> > > > - **object_type** [string] The type of the object. Valid types include: cass_ncoa, container_script, geocode, python_script, r_script, salesforce_export, javascript_script, sql_script, project, notebook, workflow, template_script, template_report, service, report, tableau and service_report.

> > > > - **alias** [string] The alias of the object

> > > > - **user_id** [integer] The id of the user who created the alias

> > • **display_name** [string] The display name of the Alias object. Defaults to
> > object name if not provided.

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

> Set the permissions groups has on this object

> > **Parameters**

> > > **id** [integer] The ID of the resource that is shared.

> > > **group_ids** [list] An array of one or more group IDs.

> > > **permission_level** [string] Options are: "read", "write", or "manage".

> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.

> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.

> > **Returns**

> > > [*civis.response.Response*](#)

> > > > • **readers** [dict::]

> > > > > – **users** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > > > – **groups** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > > • **writers** [dict::]

> > > > > – **users** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > > > – **groups** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > > • **owners** [dict::]

> > > > > – **users** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > > > – **groups** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > > • **total_user_shares** [integer] For owners, the number of total users shared.
> > > > For writers and readers, the number of visible users shared.

> > > > • **total_group_shares** [integer] For owners, the number of total groups
> > > > shared. For writers and readers, the number of visible groups shared.

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

  Set the permissions users have on this object

  > **Parameters**
  >
  > > **id** [integer] The ID of the resource that is shared.
  > >
  > > **user_ids** [list] An array of one or more user IDs.
  > >
  > > **permission_level** [string] Options are: "read", "write", or "manage".
  > >
  > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
  > >
  > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
  >
  > **Returns**
  >
  > > [*civis.response.Response*](#)
  > >
  > > > • **readers** [dict::]
  > > >
  > > > > – **users** [list::]
  > > > >
  > > > > > ∗ id : integer
  > > > > >
  > > > > > ∗ name : string
  > > > >
  > > > > – **groups** [list::]
  > > > >
  > > > > > ∗ id : integer
  > > > > >
  > > > > > ∗ name : string
  > > >
  > > > • **writers** [dict::]
  > > >
  > > > > – **users** [list::]
  > > > >
  > > > > > ∗ id : integer
  > > > > >
  > > > > > ∗ name : string
  > > > >
  > > > > – **groups** [list::]
  > > > >
  > > > > > ∗ id : integer
  > > > > >
  > > > > > ∗ name : string
  > > >
  > > > • **owners** [dict::]
  > > >
  > > > > – **users** [list::]
  > > > >
  > > > > > ∗ id : integer
  > > > > >
  > > > > > ∗ name : string
  > > > >
  > > > > – **groups** [list::]
  > > > >
  > > > > > ∗ id : integer
  > > > > >
  > > > > > ∗ name : string
  > > >
  > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
  > > >
  > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_transfer(*id*, *user_id*, *include_dependencies*, *, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

  Transfer ownership of this object to another user

  > **Parameters**

id [integer] The ID of the resource that is shared.

user_id [integer] ID of target user

include_dependencies [boolean] Whether or not to give manage permissions on all dependencies

email_body [string, optional] Custom body text for e-mail sent on transfer.

send_email [boolean, optional] Send email to the target user of the transfer?

Returns

> *civis.response.Response*
>
> > • **dependencies** [list::] Dependent objects for this object - object_type : string
> >
> > > Dependent object type
> > >
> > > – **fco_type** [string] Human readable dependent object type
> > >
> > > – **id** [integer] Dependent object ID
> > >
> > > – **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > >
> > > – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > >
> > > – **shared** [boolean] Whether dependent object was successfully shared with target user

Announcements

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

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.announcements.list(...)
```

Methods

list(*[, limit, page_num, order, order_dir, ...])	List announcements

——

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

> List announcements
>
> > **Parameters**
> >
> > > **limit** [integer, optional] Number of results to return. Defaults to 10. Maximum allowed is 50.

> **page_num** [integer, optional] Page number of the results to return. Defaults to the first
> page, 1.
>
> **order** [string, optional] The field on which to order the result set. Defaults to re-
> leased_at. Must be one of: released_at.
>
> **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc
> (descending) defaulting to desc.
>
> **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use
> when more results than the maximum allowed by limit are needed. When True,
> limit and page_num are ignored. Defaults to False.

> **Returns**
>
> > *civis.response.PaginatedResponse*
> >
> > - **id** [integer] The ID of this announcement
> >
> > - **subject** [string] The subject of this announcement.
> >
> > - **body** [string] The body of this announcement.
> >
> > - **released_at** [string/date-time] The date and time this announcement was
> >   released.
> >
> > - created_at : string/date-time
> >
> > - updated_at : string/date-time

Clusters

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

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.clusters.list_kubernetes(...)
```

Methods

delete_kubernetes_partitions(id, ...)	Delete a Cluster Partition
get_kubernetes(id, *[, include_usage_stats])	Describe a Kubernetes Cluster
get_kubernetes_instance_configs(...[, ...])	Describe an Instance Config
get_kubernetes_partitions(id, ...[, ...])	Describe a Cluster Partition
list_kubernetes(*[, organization_slug, ...])	List Kubernetes Clusters
list_kubernetes_deployment_stats(id)	Get stats about deployments associated with a Kubernetes Cluster
list_kubernetes_deployments(id, *[, ...])	List the deployments associated with a Kubernetes Cluster
list_kubernetes_instance_configs_active_workloads(id, *)	List active workloads in an Instance Config

Table 13 – continued from previous page

list_kubernetes_instance_configs_historical(id, ...)	Get graphs of historical resource usage in an Instance Config
list_kubernetes_instance_configs_user_stats(id, ...)	Get statistics about the current users of an Instance Config
list_kubernetes_partitions(id, *[, ...])	List Cluster Partitions for given cluster
patch_kubernetes(id, *[, raw_cluster_slug, ...])	Update a Kubernetes Cluster
patch_kubernetes_partitions(id, ...[, ...])	Update a Cluster Partition
post_kubernetes(*[, organization_id, ...])	Create a Kubernetes Cluster
post_kubernetes_partitions(id, ...)	Create a Cluster Partition for given cluster

—

delete_kubernetes_partitions(*id*, *cluster_partition_id*)

    Delete a Cluster Partition

        Parameters

            id [integer] The ID of the cluster which this partition belongs to.

            cluster_partition_id [integer] The ID of this cluster partition.

        Returns

            None Response code 204: success

get_kubernetes(*id*, ***, *include_usage_stats='DEFAULT'*)

    Describe a Kubernetes Cluster

        Parameters

            id [integer]

            include_usage_stats [boolean, optional] When true, usage stats are returned in instance config objects. Defaults to false.

        Returns

            civis.response.Response

                • **id** [integer] The ID of this cluster.

                • **organization_id** [string] The id of this cluster's organization.

                • **organization_name** [string] The name of this cluster's organization.

                • **organization_slug** [string] The slug of this cluster's organization.

                • **raw_cluster_slug** [string] The slug of this cluster's raw configuration.

                • **custom_partitions** [boolean] Whether this cluster has a custom partition configuration.

                • **cluster_partitions** [list::] List of cluster partitions associated with this cluster. - cluster_partition_id : integer

                    The ID of this cluster partition.

                  – **name** [string] The name of the cluster partition.

                  – **labels** [list] Labels associated with this partition.

                  – **instance_configs** [list::] The instances configured for this cluster partition. - instance_config_id : integer

                    The ID of this InstanceConfig.

* **instance_type** [string] An EC2 instance type. Possible values include t2.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m5.12xlarge, c5.18xlarge, and p2.xlarge.

* **min_instances** [integer] The minimum number of instances of that type in this cluster.

* **max_instances** [integer] The maximum number of instances of that type in this cluster.

* **instance_max_memory** [integer] The amount of memory (RAM) available to a single instance of that type in megabytes.

* **instance_max_cpu** [integer] The number of processor shares available to a single instance of that type in millicores.

* **instance_max_disk** [integer] The amount of disk available to a single instance of that type in gigabytes.

* **usage_stats** [dict::]

  · **pending_memory_requested** [integer] The sum of memory requests (in MB) for pending deployments in this instance config.

  · **pending_cpu_requested** [integer] The sum of cpu requests (in millicores) for pending deployments in this instance config.

  · **running_memory_requested** [integer] The sum of memory requests (in MB) for running deployments in this instance config.

  · **running_cpu_requested** [integer] The sum of cpu requests (in millicores) for running deployments in this instance config.

  · **pending_deployments** [integer] The number of pending deployments in this instance config.

  · **running_deployments** [integer] The number of running deployments in this instance config.

– **default_instance_config_id** [integer] The id of the InstanceConfig that is the default for this partition.

• **is_nat_enabled** [boolean] Whether this cluster needs a NAT gateway or not.

• **hours** [number/float] The number of hours used this month for this cluster.

get_kubernetes_instance_configs(*instance_config_id*, *, *include_usage_stats='DEFAULT'*)

>   Describe an Instance Config

>   **Parameters**

>>   **instance_config_id** [integer] The ID of this instance config.

>>   **include_usage_stats** [boolean, optional] When true, usage stats are returned in instance config objects. Defaults to false.

>   **Returns**

>>   *civis.response.Response*

- **instance_config_id** [integer] The ID of this InstanceConfig.
- **instance_type** [string] An EC2 instance type. Possible values include t2.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m5.12xlarge, c5.18xlarge, and p2.xlarge.
- **min_instances** [integer] The minimum number of instances of that type in this cluster.
- **max_instances** [integer] The maximum number of instances of that type in this cluster.
- **instance_max_memory** [integer] The amount of memory (RAM) available to a single instance of that type in megabytes.
- **instance_max_cpu** [integer] The number of processor shares available to a single instance of that type in millicores.
- **instance_max_disk** [integer] The amount of disk available to a single instance of that type in gigabytes.
- **usage_stats** [dict::]

    – **pending_memory_requested** [integer] The sum of memory requests (in MB) for pending deployments in this instance config.

    – **pending_cpu_requested** [integer] The sum of cpu requests (in millicores) for pending deployments in this instance config.

    – **running_memory_requested** [integer] The sum of memory requests (in MB) for running deployments in this instance config.

    – **running_cpu_requested** [integer] The sum of cpu requests (in millicores) for running deployments in this instance config.

    – **pending_deployments** [integer] The number of pending deployments in this instance config.

    – **running_deployments** [integer] The number of running deployments in this instance config.

- **cluster_partition_id** [integer] The ID of this InstanceConfig's cluster partition
- **cluster_partition_name** [string] The name of this InstanceConfig's cluster partition

get_kubernetes_partitions(*id*, *cluster_partition_id*, *, *include_usage_stats='DEFAULT'*)

Describe a Cluster Partition

Parameters

   id [integer] The ID of the cluster which this partition belongs to.

   cluster_partition_id [integer] The ID of this cluster partition.

   include_usage_stats [boolean, optional] When true, usage stats are returned in instance config objects. Defaults to false.

Returns

civis.response.Response

- **cluster_partition_id** [integer] The ID of this cluster partition.
- **name** [string] The name of the cluster partition.
- **labels** [list] Labels associated with this partition.
- **instance_configs** [list::] The instances configured for this cluster partition. - instance_config_id : integer

    The ID of this InstanceConfig.

> > > > – **instance_type** [string] An EC2 instance type. Possible values include t2.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m5.12xlarge, c5.18xlarge, and p2.xlarge.
> > > >
> > > > – **min_instances** [integer] The minimum number of instances of that type in this cluster.
> > > >
> > > > – **max_instances** [integer] The maximum number of instances of that type in this cluster.
> > > >
> > > > – **instance_max_memory** [integer] The amount of memory (RAM) available to a single instance of that type in megabytes.
> > > >
> > > > – **instance_max_cpu** [integer] The number of processor shares available to a single instance of that type in millicores.
> > > >
> > > > – **instance_max_disk** [integer] The amount of disk available to a single instance of that type in gigabytes.
> > > >
> > > > – **usage_stats** [dict::]
> > > >
> > > > > * **pending_memory_requested** [integer] The sum of memory requests (in MB) for pending deployments in this instance config.
> > > > >
> > > > > * **pending_cpu_requested** [integer] The sum of cpu requests (in millicores) for pending deployments in this instance config.
> > > > >
> > > > > * **running_memory_requested** [integer] The sum of memory requests (in MB) for running deployments in this instance config.
> > > > >
> > > > > * **running_cpu_requested** [integer] The sum of cpu requests (in millicores) for running deployments in this instance config.
> > > > >
> > > > > * **pending_deployments** [integer] The number of pending deployments in this instance config.
> > > > >
> > > > > * **running_deployments** [integer] The number of running deployments in this instance config.
> > >
> > > • **default_instance_config_id** [integer] The id of the InstanceConfig that is the default for this partition.

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

> List Kubernetes Clusters
>
> > **Parameters**
> >
> > > **organization_slug** [string, optional] The slug of this cluster's organization.
> > >
> > > **raw_cluster_slug** [string, optional] The slug of this cluster's raw configuration.
> > >
> > > **exclude_inactive_orgs** [boolean, optional] When true, excludes KubeClusters associated with inactive orgs. Defaults to false.
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to its maximum of 50.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to organization_id. Must be one of: organization_id, created_at.

order_dir [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to asc.

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

Returns

civis.response.PaginatedResponse

- **id** [integer] The ID of this cluster.
- **organization_id** [string] The id of this cluster's organization.
- **organization_name** [string] The name of this cluster's organization.
- **organization_slug** [string] The slug of this cluster's organization.
- **raw_cluster_slug** [string] The slug of this cluster's raw configuration.
- **custom_partitions** [boolean] Whether this cluster has a custom partition configuration.
- **cluster_partitions** [list::] List of cluster partitions associated with this cluster. - cluster_partition_id : integer

  The ID of this cluster partition.

  - **name** [string] The name of the cluster partition.

  - **labels** [list] Labels associated with this partition.

  - **instance_configs** [list::] The instances configured for this cluster partition. - instance_config_id : integer

    The ID of this InstanceConfig.

    * **instance_type** [string] An EC2 instance type. Possible values include t2.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m5.12xlarge, c5.18xlarge, and p2.xlarge.

    * **min_instances** [integer] The minimum number of instances of that type in this cluster.

    * **max_instances** [integer] The maximum number of instances of that type in this cluster.

    * **instance_max_memory** [integer] The amount of memory (RAM) available to a single instance of that type in megabytes.

    * **instance_max_cpu** [integer] The number of processor shares available to a single instance of that type in millicores.

    * **instance_max_disk** [integer] The amount of disk available to a single instance of that type in gigabytes.

    * **usage_stats** [dict::]

      · **pending_memory_requested** [integer] The sum of memory requests (in MB) for pending deployments in this instance config.

> > > · **pending_cpu_requested** [integer] The sum of cpu requests (in millicores) for pending deployments in this instance config.
> > >
> > > · **running_memory_requested** [integer] The sum of memory requests (in MB) for running deployments in this instance config.
> > >
> > > · **running_cpu_requested** [integer] The sum of cpu requests (in millicores) for running deployments in this instance config.
> > >
> > > · **pending_deployments** [integer] The number of pending deployments in this instance config.
> > >
> > > · **running_deployments** [integer] The number of running deployments in this instance config.
> >
> > – **default_instance_config_id** [integer] The id of the Instance-Config that is the default for this partition.
>
> • **is_nat_enabled** [boolean] Whether this cluster needs a NAT gateway or not.

list_kubernetes_deployment_stats(*id*)

> Get stats about deployments associated with a Kubernetes Cluster
>
> > **Parameters**
> >
> > > **id** [integer] The ID of this cluster.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **base_type** [string] The base type of this deployment
> > > > • **state** [string] State of the deployment
> > > > • **count** [integer] Number of deployments of base type and state
> > > > • **total_cpu** [integer] Total amount of CPU in millicores for deployments of base type and state
> > > > • **total_memory** [integer] Total amount of Memory in megabytes for deployments of base type and state

list_kubernetes_deployments(*id, *, base_type='DEFAULT', state='DEFAULT', limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT', iterator='DEFAULT'*)

> List the deployments associated with a Kubernetes Cluster
>
> > **Parameters**
> >
> > > **id** [integer] The id of the cluster.
> > >
> > > **base_type** [string, optional] If specified, return deployments of these base types. It accepts a comma- separated list, possible values are 'Notebook', 'Service', 'Run'.
> > >
> > > **state** [string, optional] If specified, return deployments in these states. It accepts a comma- separated list, possible values are pending, running, terminated, sleeping
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to its maximum of 50.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to created_at. Must be one of: created_at.

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

civis.response.PaginatedResponse

- **id** [integer] The id of this deployment.
- **name** [string] The name of the deployment.
- **base_id** [integer] The id of the base object associated with the deployment.
- **base_type** [string] The base type of this deployment.
- **state** [string] The state of the deployment.
- **cpu** [integer] The CPU in millicores required by the deployment.
- **memory** [integer] The memory in MB required by the deployment.
- **disk_space** [integer] The disk space in GB required by the deployment.
- **instance_type** [string] The EC2 instance type requested for the deployment.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.
- **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.
- created_at : string/time
- updated_at : string/time

list_kubernetes_instance_configs_active_workloads(*id*, ***, *state='DEFAULT'*)

List active workloads in an Instance Config

Parameters

id [integer] The id of the instance config.

state [string, optional] If specified, return workloads in these states. It accepts a comma- separated list, possible values are pending, running

Returns

civis.response.Response

- **id** [integer] The id of this deployment.
- **base_type** [string] The base type of this deployment.
- **base_id** [integer] The id of the base object associated with this deployment.
- **base_object_name** [string] The name of the base object associated with this deployment. Null if you do not have permission to read the object.
- **job_type** [string] If the base object is a job run you have permission to read, the type of the job. One of "python_script", "r_script", "container_script", or "custom_script".
- **job_id** [integer] If the base object is a job run you have permission to read, the id of the job.
- **job_cancel_requested_at** [string/time] If the base object is a job run you have permission to read, and it was requested to be cancelled, the

> timestamp of that request.
> - **state** [string] The state of this deployment.
> - **cpu** [integer] The CPU in millicores requested by this deployment.
> - **memory** [integer] The memory in MB requested by this deployment.
> - **disk_space** [integer] The disk space in GB requested by this deployment.
> - **user** [dict::]
>
>> - **id** [integer] The ID of this user.
>>
>> - **name** [string] This user's name.
>>
>> - **username** [string] This user's username.
>>
>> - **initials** [string] This user's initials.
>>
>> - **online** [boolean] Whether this user is online.
> - **created_at** [string/time] The timestamp of when the deployment began.
> - **cancellable** [boolean] True if you have permission to cancel this deployment.

list_kubernetes_instance_configs_historical_graphs(*instance_config_id*, ***, *timeframe='DEFAULT'*)

> Get graphs of historical resource usage in an Instance Config
>
> **Parameters**
>> **instance_config_id** [integer] The ID of this instance config.
>>
>> **timeframe** [string, optional] The span of time that the graphs cover. Must be one of 1_day, 1_week.
>
> **Returns**
>> `civis.response.Response`
>>
>>> - **cpu_graph_url** [string] URL for the graph of historical CPU usage in this instance config.
>>> - **mem_graph_url** [string] URL for the graph of historical memory usage in this instance config.

list_kubernetes_instance_configs_user_statistics(*instance_config_id*, ***, *order='DEFAULT'*, *order_dir='DEFAULT'*)

> Get statistics about the current users of an Instance Config
>
> **Parameters**
>> **instance_config_id** [integer] The ID of this instance config.
>>
>> **order** [string, optional] The field on which to order the result set. Defaults to running_deployments. Must be one of pending_memory_requested, pending_cpu_requested, running_memory_requested, running_cpu_requested, pending_deployments, running_deployments.
>>
>> **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending). Defaults to desc.
>
> **Returns**
>> `civis.response.Response`
>>
>>> - **user_id** [string] The owning user's ID
>>> - **user_name** [string] The owning user's name
>>> - **pending_deployments** [integer] The number of deployments belonging to the owning user in "pending" state
>>> - **pending_memory_requested** [integer] The sum of memory requests (in MB) for deployments belonging to the owning user in "pending" state
>>> - **pending_cpu_requested** [integer] The sum of CPU requests (in millicores) for deployments belonging to the owning user in "pending" state

- **running_deployments** [integer] The number of deployments belonging to the owning user in "running" state
- **running_memory_requested** [integer] The sum of memory requests (in MB) for deployments belonging to the owning user in "running" state
- **running_cpu_requested** [integer] The sum of CPU requests (in millicores) for deployments belonging to the owning user in "running" state

list_kubernetes_partitions(*id*, ***, *include_usage_stats='DEFAULT'*)
List Cluster Partitions for given cluster

> **Parameters**
> > **id** [integer]
> > **include_usage_stats** [boolean, optional] When true, usage stats are returned in instance config objects. Defaults to false.
>
> **Returns**
> > *civis.response.Response*
> > - **cluster_partition_id** [integer] The ID of this cluster partition.
> > - **name** [string] The name of the cluster partition.
> > - **labels** [list] Labels associated with this partition.
> > - **instance_configs** [list::] The instances configured for this cluster partition. - instance_config_id : integer
> >
> > > The ID of this InstanceConfig.
> > >
> > > – **instance_type** [string] An EC2 instance type. Possible values include t2.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m5.12xlarge, c5.18xlarge, and p2.xlarge.
> > >
> > > – **min_instances** [integer] The minimum number of instances of that type in this cluster.
> > >
> > > – **max_instances** [integer] The maximum number of instances of that type in this cluster.
> > >
> > > – **instance_max_memory** [integer] The amount of memory (RAM) available to a single instance of that type in megabytes.
> > >
> > > – **instance_max_cpu** [integer] The number of processor shares available to a single instance of that type in millicores.
> > >
> > > – **instance_max_disk** [integer] The amount of disk available to a single instance of that type in gigabytes.
> > >
> > > – **usage_stats** [dict::]
> > >
> > > > * **pending_memory_requested** [integer] The sum of memory requests (in MB) for pending deployments in this instance config.
> > > >
> > > > * **pending_cpu_requested** [integer] The sum of cpu requests (in millicores) for pending deployments in this instance config.
> > > >
> > > > * **running_memory_requested** [integer] The sum of memory requests (in MB) for running deployments in this instance config.

---

* **running_cpu_requested** [integer] The sum of cpu requests (in millicores) for running deployments in this instance config.

* **pending_deployments** [integer] The number of pending deployments in this instance config.

* **running_deployments** [integer] The number of running deployments in this instance config.

- **default_instance_config_id** [integer] The id of the InstanceConfig that is the default for this partition.

patch_kubernetes(*id*, ***, *raw_cluster_slug='DEFAULT'*, *is_nat_enabled='DEFAULT'*)

Update a Kubernetes Cluster

> **Parameters**
>> **id** [integer] The ID of this cluster.
>> **raw_cluster_slug** [string, optional] The slug of this cluster's raw configuration.
>> **is_nat_enabled** [boolean, optional] Whether this cluster needs a NAT gateway or not.
>
> **Returns**
>> [*civis.response.Response*](#)
>>> - **id** [integer] The ID of this cluster.
>>> - **organization_id** [string] The id of this cluster's organization.
>>> - **organization_name** [string] The name of this cluster's organization.
>>> - **organization_slug** [string] The slug of this cluster's organization.
>>> - **raw_cluster_slug** [string] The slug of this cluster's raw configuration.
>>> - **custom_partitions** [boolean] Whether this cluster has a custom partition configuration.
>>> - **cluster_partitions** [list::] List of cluster partitions associated with this cluster. - cluster_partition_id : integer
>>>
>>>> The ID of this cluster partition.
>>>
>>>> – **name** [string] The name of the cluster partition.
>>>
>>>> – **labels** [list] Labels associated with this partition.
>>>
>>>> – **instance_configs** [list::] The instances configured for this cluster partition. - instance_config_id : integer
>>>>
>>>>> The ID of this InstanceConfig.
>>>>
>>>>> * **instance_type** [string] An EC2 instance type. Possible values include t2.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m5.12xlarge, c5.18xlarge, and p2.xlarge.
>>>>
>>>>> * **min_instances** [integer] The minimum number of instances of that type in this cluster.
>>>>
>>>>> * **max_instances** [integer] The maximum number of instances of that type in this cluster.
>>>>
>>>>> * **instance_max_memory** [integer] The amount of memory (RAM) available to a single instance of that type in megabytes.
>>>>
>>>>> * **instance_max_cpu** [integer] The number of processor shares available to a single instance of that type in millicores.

        ∗ **instance_max_disk** [integer] The amount of disk available to a single instance of that type in gigabytes.

        ∗ **usage_stats** [dict::]

           · **pending_memory_requested** [integer] The sum of memory requests (in MB) for pending deployments in this instance config.

           · **pending_cpu_requested** [integer] The sum of cpu requests (in millicores) for pending deployments in this instance config.

           · **running_memory_requested** [integer] The sum of memory requests (in MB) for running deployments in this instance config.

           · **running_cpu_requested** [integer] The sum of cpu requests (in millicores) for running deployments in this instance config.

           · **pending_deployments** [integer] The number of pending deployments in this instance config.

           · **running_deployments** [integer] The number of running deployments in this instance config.

      – **default_instance_config_id** [integer] The id of the InstanceConfig that is the default for this partition.

- **is_nat_enabled** [boolean] Whether this cluster needs a NAT gateway or not.
- **hours** [number/float] The number of hours used this month for this cluster.

patch_kubernetes_partitions(*id*, *cluster_partition_id*, ***, *instance_configs='DEFAULT'*, *name='DEFAULT'*, *labels='DEFAULT'*)

    Update a Cluster Partition

        Parameters

                id [integer] The ID of the cluster which this partition belongs to.

                cluster_partition_id [integer] The ID of this cluster partition.

                instance_configs [list, optional::] The instances configured for this cluster partition. - instance_type : string

                    An EC2 instance type. Possible values include t2.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m5.12xlarge, c5.18xlarge, and p2.xlarge.

                    - **min_instances** [integer] The minimum number of instances of that type in this cluster.
                    - **max_instances** [integer] The maximum number of instances of that type in this cluster.

                name [string, optional] The name of the cluster partition.

                labels [list, optional] Labels associated with this partition.

        Returns

            civis.response.Response

                - **cluster_partition_id** [integer] The ID of this cluster partition.

- **name** [string] The name of the cluster partition.
- **labels** [list] Labels associated with this partition.
- **instance_configs** [list::] The instances configured for this cluster partition. - instance_config_id : integer

    The ID of this InstanceConfig.

    - **instance_type** [string] An EC2 instance type. Possible values include t2.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m5.12xlarge, c5.18xlarge, and p2.xlarge.

    - **min_instances** [integer] The minimum number of instances of that type in this cluster.

    - **max_instances** [integer] The maximum number of instances of that type in this cluster.

    - **instance_max_memory** [integer] The amount of memory (RAM) available to a single instance of that type in megabytes.

    - **instance_max_cpu** [integer] The number of processor shares available to a single instance of that type in millicores.

    - **instance_max_disk** [integer] The amount of disk available to a single instance of that type in gigabytes.

    - **usage_stats** [dict::]

        * **pending_memory_requested** [integer] The sum of memory requests (in MB) for pending deployments in this instance config.

        * **pending_cpu_requested** [integer] The sum of cpu requests (in millicores) for pending deployments in this instance config.

        * **running_memory_requested** [integer] The sum of memory requests (in MB) for running deployments in this instance config.

        * **running_cpu_requested** [integer] The sum of cpu requests (in millicores) for running deployments in this instance config.

        * **pending_deployments** [integer] The number of pending deployments in this instance config.

        * **running_deployments** [integer] The number of running deployments in this instance config.

- **default_instance_config_id** [integer] The id of the InstanceConfig that is the default for this partition.

post_kubernetes(*, *organization_id='DEFAULT'*, *organization_slug='DEFAULT'*, *raw_cluster_slug='DEFAULT'*, *is_nat_enabled='DEFAULT'*)

   Create a Kubernetes Cluster

   Parameters

   organization_id [string, optional] The id of this cluster's organization.

   organization_slug [string, optional] The slug of this cluster's organization.

   raw_cluster_slug [string, optional] The slug of this cluster's raw configuration.

> **is_nat_enabled** [boolean, optional] Whether this cluster needs a NAT gateway or not.

Returns

civis.response.Response

- **id** [integer] The ID of this cluster.
- **organization_id** [string] The id of this cluster's organization.
- **organization_name** [string] The name of this cluster's organization.
- **organization_slug** [string] The slug of this cluster's organization.
- **raw_cluster_slug** [string] The slug of this cluster's raw configuration.
- **custom_partitions** [boolean] Whether this cluster has a custom partition configuration.
- **cluster_partitions** [list::] List of cluster partitions associated with this cluster. - cluster_partition_id : integer

    The ID of this cluster partition.

    – **name** [string] The name of the cluster partition.

    – **labels** [list] Labels associated with this partition.

    – **instance_configs** [list::] The instances configured for this cluster partition. - instance_config_id : integer

        The ID of this InstanceConfig.

        * **instance_type** [string] An EC2 instance type. Possible values include t2.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m5.12xlarge, c5.18xlarge, and p2.xlarge.

        * **min_instances** [integer] The minimum number of instances of that type in this cluster.

        * **max_instances** [integer] The maximum number of instances of that type in this cluster.

        * **instance_max_memory** [integer] The amount of memory (RAM) available to a single instance of that type in megabytes.

        * **instance_max_cpu** [integer] The number of processor shares available to a single instance of that type in millicores.

        * **instance_max_disk** [integer] The amount of disk available to a single instance of that type in gigabytes.

        * **usage_stats** [dict::]

            · **pending_memory_requested** [integer] The sum of memory requests (in MB) for pending deployments in this instance config.

            · **pending_cpu_requested** [integer] The sum of cpu requests (in millicores) for pending deployments in this instance config.

            · **running_memory_requested** [integer] The sum of memory requests (in MB)

> > > > > for running deployments in this instance config.

> > > > · **running_cpu_requested** [integer] The sum of cpu requests (in millicores) for running deployments in this instance config.

> > > > · **pending_deployments** [integer] The number of pending deployments in this instance config.

> > > > · **running_deployments** [integer] The number of running deployments in this instance config.

> > > – **default_instance_config_id** [integer] The id of the Instance-Config that is the default for this partition.

> > • **is_nat_enabled** [boolean] Whether this cluster needs a NAT gateway or not.
> > • **hours** [number/float] The number of hours used this month for this cluster.

post_kubernetes_partitions(*id*, *instance_configs*, *name*, *labels*)

> Create a Cluster Partition for given cluster

> > **Parameters**

> > > **id** [integer] The ID of the cluster which this partition belongs to.

> > > **instance_configs** [list::] The instances configured for this cluster partition. - instance_type : string

> > > > An EC2 instance type. Possible values include t2.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m5.12xlarge, c5.18xlarge, and p2.xlarge.

> > > > • **min_instances** [integer] The minimum number of instances of that type in this cluster.
> > > > • **max_instances** [integer] The maximum number of instances of that type in this cluster.

> > > **name** [string] The name of the cluster partition.
> > > **labels** [list] Labels associated with this partition.

> > **Returns**

> > > *civis.response.Response*

> > > > • **cluster_partition_id** [integer] The ID of this cluster partition.
> > > > • **name** [string] The name of the cluster partition.
> > > > • **labels** [list] Labels associated with this partition.
> > > > • **instance_configs** [list::] The instances configured for this cluster partition. - instance_config_id : integer

> > > > > The ID of this InstanceConfig.

> > > > – **instance_type** [string] An EC2 instance type. Possible values include t2.large, m4.xlarge, m4.2xlarge, m4.4xlarge, m5.12xlarge, c5.18xlarge, and p2.xlarge.

> > > > – **min_instances** [integer] The minimum number of instances of that type in this cluster.

> > > > – **max_instances** [integer] The maximum number of instances of that type in this cluster.

> > > > – **instance_max_memory** [integer] The amount of memory (RAM) available to a single instance of that type in

> megabytes.
>
>> – **instance_max_cpu** [integer] The number of processor shares available to a single instance of that type in millicores.
>>
>> – **instance_max_disk** [integer] The amount of disk available to a single instance of that type in gigabytes.
>>
>> – **usage_stats** [dict::]
>>
>>> * **pending_memory_requested** [integer] The sum of memory requests (in MB) for pending deployments in this instance config.
>>>
>>> * **pending_cpu_requested** [integer] The sum of cpu requests (in millicores) for pending deployments in this instance config.
>>>
>>> * **running_memory_requested** [integer] The sum of memory requests (in MB) for running deployments in this instance config.
>>>
>>> * **running_cpu_requested** [integer] The sum of cpu requests (in millicores) for running deployments in this instance config.
>>>
>>> * **pending_deployments** [integer] The number of pending deployments in this instance config.
>>>
>>> * **running_deployments** [integer] The number of running deployments in this instance config.
>
> • **default_instance_config_id** [integer] The id of the InstanceConfig that is the default for this partition.

Credentials

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

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.credentials.list_types(...)
```

Methods

delete(id)	Delete a credential
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 a credential
list(*[, type, remote_host_id, default, ...])	List credentials
list_dependencies(id, *[, user_id])	List dependent objects for this object

continues on next page

Table 15 – continued from previous page

`list_shares`(id)	List users and groups permissioned on this object
`list_types`()	Get list of Credential Types
`patch`(id, *[, name, type, description, ...])	Update some attributes of a credential
`post`(type, username, password, *[, name, ...])	Create a credential
`post_authenticate`(url, remote_host_type, ...)	Authenticate against a remote host
`post_temporary`(id, *[, duration])	Generate a temporary credential for accessing S3
`put`(id, type, username, password, *[, name, ...])	Update an existing credential
`put_shares_groups`(id, group_ids, ...[, ...])	Set the permissions groups has on this object
`put_shares_users`(id, user_ids, ...[, ...])	Set the permissions users have on this object
`put_transfer`(id, user_id, ...[, email_body, ...])	Transfer ownership of this object to another user

——

delete(*id*)

> Delete a credential
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the credential.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

delete_shares_groups(*id*, *group_id*)

> Revoke the permissions a group has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **group_id** [integer] The 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] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] The ID of the user.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

get(*id*)

> Get a credential
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the credential.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **id** [integer] The ID of the credential.
> > > >
> > > > • **name** [string] The name identifying the credential
> > > >
> > > > • **type** [string] The credential's type.
> > > >
> > > > • **username** [string] The username for the credential.
> > > >
> > > > • **description** [string] A long description of the credential.
> > > >
> > > > • **owner** [string] The username of the user who this credential belongs to.
> > > >   Using user.username is preferred.
> > > >
> > > > • **user** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.

- **remote_host_id** [integer] The ID of the remote host associated with this credential.

- **remote_host_name** [string] The name of the remote host associated with this credential.

- **state** [string] The U.S. state for the credential. Only for VAN credentials.

- **created_at** [string/time] The creation time for this credential.

- **updated_at** [string/time] The last modification time for this credential.

- **default** [boolean] Whether or not the credential is a default. Only for Database credentials.

list(*, *type='DEFAULT'*, *remote_host_id='DEFAULT'*, *default='DEFAULT'*, *system_credentials='DEFAULT'*, *users='DEFAULT'*, *name='DEFAULT'*, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)

List credentials

> **Parameters**
>> **type** [string, optional] The type (or types) of credentials to return. One or more of: Amazon Web Services S3, Bitbucket, CASS/NCOA PAF, Certificate, Civis Platform, Custom, Database, Google, Github, Salesforce User, Salesforce Client, and TableauUser. Specify multiple values as a comma-separated list (e.g., "A,B").
>>
>> **remote_host_id** [integer, optional] The ID of the remote host associated with the credentials to return.
>>
>> **default** [boolean, optional] If true, will return a list with a single credential which is the current user's default credential.
>>
>> **system_credentials** [boolean, optional] If true, will only return system credentials. System credentials can only be created and viewed by Civis Admins.
>>
>> **users** [string, optional] A comma-separated list of user ids. If specified, returns set of credentials owned by the users that requesting user has at least read access on.
>>
>> **name** [string, optional] If specified, will be used to filter the credentials returned. Will search across name and will return any full name containing the search string.
>>
>> **limit** [integer, optional] Number of results to return. Defaults to its maximum of 1000.
>>
>> **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
>>
>> **order** [string, optional] The field on which to order the result set. Defaults to updated_at. Must be one of: updated_at, created_at, name.
>>
>> **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
>>
>> **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
>
> **Returns**
>> *civis.response.PaginatedResponse*
>>> - **id** [integer] The ID of the credential.
>>> - **name** [string] The name identifying the credential
>>> - **type** [string] The credential's type.
>>> - **username** [string] The username for the credential.
>>> - **description** [string] A long description of the credential.
>>> - **owner** [string] The username of the user who this credential belongs to. Using user.username is preferred.
>>> - **user** [dict::]
>>>
>>>> - **id** [integer] The ID of this user.

> > > – **name** [string] This user's name.
> > >
> > > – **username** [string] This user's username.
> > >
> > > – **initials** [string] This user's initials.
> > >
> > > – **online** [boolean] Whether this user is online.
> >
> > • **remote_host_id** [integer] The ID of the remote host associated with this credential.
> > • **remote_host_name** [string] The name of the remote host associated with this credential.
> > • **state** [string] The U.S. state for the credential. Only for VAN credentials.
> > • **created_at** [string/time] The creation time for this credential.
> > • **updated_at** [string/time] The last modification time for this credential.
> > • **default** [boolean] Whether or not the credential is a default. Only for Database credentials.

list_dependencies(*id*, ***, *user_id='DEFAULT'*)

> List dependent objects for this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_id** [integer, optional] ID of target user
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **object_type** [string] Dependent object type
> > > > • **fco_type** [string] Human readable dependent object type
> > > > • **id** [integer] Dependent object ID
> > > > • **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > > > • **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > > > • **shareable** [boolean] Whether or not the requesting user can share this object.

list_shares(*id*)

> List users and groups permissioned on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **readers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **writers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string

- **groups** [list::]

  ∗ id : integer

  ∗ name : string

- **owners** [dict::]

  – **users** [list::]

    ∗ id : integer

    ∗ name : string

  – **groups** [list::]

    ∗ id : integer

    ∗ name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_types()

> Get list of Credential Types
>
> > **Returns**
> >
> > > [`civis.response.Response`](#)
> > >
> > > > - **types** [list] list of acceptable credential types

patch(*id*, ***, *name='DEFAULT'*, *type='DEFAULT'*, *description='DEFAULT'*, *username='DEFAULT'*, *password='DEFAULT'*, *remote_host_id='DEFAULT'*, *user_id='DEFAULT'*, *state='DEFAULT'*, *system_credential='DEFAULT'*, *default='DEFAULT'*)

> Update some attributes of a credential
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the credential.
> > >
> > > **name** [string, optional] The name identifying the credential.
> > >
> > > **type** [string, optional] The type of credential. Note: only these credentials can be created or edited via this API ["Amazon Web Services S3", "CASS/NCOA PAF", "Certificate", "Civis Platform", "Custom", "Database", "Google", "Salesforce User", "Salesforce Client", "TableauUser"]
> > >
> > > **description** [string, optional] A long description of the credential.
> > >
> > > **username** [string, optional] The username for the credential.
> > >
> > > **password** [string, optional] The password for the credential.
> > >
> > > **remote_host_id** [integer, optional] The ID of the remote host associated with the credential.
> > >
> > > **user_id** [integer, optional] The ID of the user the credential is created for. Note: This attribute is only accepted if you are a Civis Admin User.
> > >
> > > **state** [string, optional] The U.S. state for the credential. Only for VAN credentials.
> > >
> > > **system_credential** [boolean, optional] Boolean flag that sets a credential to be a system credential. System credentials can only be created by Civis Admins and will create a credential owned by the Civis Robot user.
> > >
> > > **default** [boolean, optional] Whether or not the credential is a default. Only for Database credentials.
> >
> > **Returns**
> >
> > > [`civis.response.Response`](#)
> > >
> > > > - **id** [integer] The ID of the credential.
> > > > - **name** [string] The name identifying the credential
> > > > - **type** [string] The credential's type.
> > > > - **username** [string] The username for the credential.

- **description** [string] A long description of the credential.
- **owner** [string] The username of the user who this credential belongs to. Using user.username is preferred.
- **user** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- **remote_host_id** [integer] The ID of the remote host associated with this credential.
- **remote_host_name** [string] The name of the remote host associated with this credential.
- **state** [string] The U.S. state for the credential. Only for VAN credentials.
- **created_at** [string/time] The creation time for this credential.
- **updated_at** [string/time] The last modification time for this credential.
- **default** [boolean] Whether or not the credential is a default. Only for Database credentials.

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

Create a credential

Parameters

type [string] The type of credential. Note: only these credentials can be created or edited via this API ["Amazon Web Services S3", "CASS/NCOA PAF", "Certificate", "Civis Platform", "Custom", "Database", "Google", "Salesforce User", "Salesforce Client", "TableauUser"]

username [string] The username for the credential.

password [string] The password for the credential.

name [string, optional] The name identifying the credential.

description [string, optional] A long description of the credential.

remote_host_id [integer, optional] The ID of the remote host associated with the credential.

user_id [integer, optional] The ID of the user the credential is created for. Note: This attribute is only accepted if you are a Civis Admin User.

state [string, optional] The U.S. state for the credential. Only for VAN credentials.

system_credential [boolean, optional] Boolean flag that sets a credential to be a system credential. System credentials can only be created by Civis Admins and will create a credential owned by the Civis Robot user.

default [boolean, optional] Whether or not the credential is a default. Only for Database credentials.

Returns

civis.response.Response

- **id** [integer] The ID of the credential.
- **name** [string] The name identifying the credential
- **type** [string] The credential's type.
- **username** [string] The username for the credential.
- **description** [string] A long description of the credential.
- **owner** [string] The username of the user who this credential belongs to. Using user.username is preferred.
- **user** [dict::]

> > – **id** [integer] The ID of this user.
>
> > – **name** [string] This user's name.
>
> > – **username** [string] This user's username.
>
> > – **initials** [string] This user's initials.
>
> > – **online** [boolean] Whether this user is online.
>
> - **remote_host_id** [integer] The ID of the remote host associated with this credential.
> - **remote_host_name** [string] The name of the remote host associated with this credential.
> - **state** [string] The U.S. state for the credential. Only for VAN credentials.
> - **created_at** [string/time] The creation time for this credential.
> - **updated_at** [string/time] The last modification time for this credential.
> - **default** [boolean] Whether or not the credential is a default. Only for Database credentials.

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

> Authenticate against a remote host
>
> > **Parameters**
> >
> > > **url** [string] The URL to your host.
> > >
> > > **remote_host_type** [string] The type of remote host. One of: RemoteHostTypes::Bigquery, RemoteHostTypes::Bitbucket, RemoteHostTypes::GitSSH, RemoteHostTypes::Github, RemoteHostTypes::GoogleDoc, RemoteHostTypes::JDBC, RemoteHostTypes::Postgres, RemoteHostTypes::Redshift, RemoteHostTypes::S3Storage, and RemoteHostTypes::Salesforce
> > >
> > > **username** [string] The username for the credential.
> > >
> > > **password** [string] The password for the credential.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID of the credential.
> > > - **name** [string] The name identifying the credential
> > > - **type** [string] The credential's type.
> > > - **username** [string] The username for the credential.
> > > - **description** [string] A long description of the credential.
> > > - **owner** [string] The username of the user who this credential belongs to. Using user.username is preferred.
> > > - **user** [dict::]
> > >
> > > > – **id** [integer] The ID of this user.
> > > >
> > > > – **name** [string] This user's name.
> > > >
> > > > – **username** [string] This user's username.
> > > >
> > > > – **initials** [string] This user's initials.
> > > >
> > > > – **online** [boolean] Whether this user is online.
> > >
> > > - **remote_host_id** [integer] The ID of the remote host associated with this credential.
> > > - **remote_host_name** [string] The name of the remote host associated with this credential.
> > > - **state** [string] The U.S. state for the credential. Only for VAN credentials.
> > > - **created_at** [string/time] The creation time for this credential.
> > > - **updated_at** [string/time] The last modification time for this credential.
> > > - **default** [boolean] Whether or not the credential is a default. Only for Database credentials.

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

> Generate a temporary credential for accessing S3
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the credential.
> > >
> > > **duration** [integer, optional] The number of seconds the temporary credential should be valid. Defaults to 15 minutes. Must not be less than 15 minutes or greater than 36 hours.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > • **access_key** [string] The identifier of the credential.
> > > • **secret_access_key** [string] The secret part of the credential.
> > > • **session_token** [string] The session token identifier.

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

> Update an existing credential
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the credential.
> > >
> > > **type** [string] The type of credential. Note: only these credentials can be created or edited via this API ["Amazon Web Services S3", "CASS/NCOA PAF", "Certificate", "Civis Platform", "Custom", "Database", "Google", "Salesforce User", "Salesforce Client", "TableauUser"]
> > >
> > > **username** [string] The username for the credential.
> > >
> > > **password** [string] The password for the credential.
> > >
> > > **name** [string, optional] The name identifying the credential.
> > >
> > > **description** [string, optional] A long description of the credential.
> > >
> > > **remote_host_id** [integer, optional] The ID of the remote host associated with the credential.
> > >
> > > **user_id** [integer, optional] The ID of the user the credential is created for. Note: This attribute is only accepted if you are a Civis Admin User.
> > >
> > > **state** [string, optional] The U.S. state for the credential. Only for VAN credentials.
> > >
> > > **system_credential** [boolean, optional] Boolean flag that sets a credential to be a system credential. System credentials can only be created by Civis Admins and will create a credential owned by the Civis Robot user.
> > >
> > > **default** [boolean, optional] Whether or not the credential is a default. Only for Database credentials.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > • **id** [integer] The ID of the credential.
> > > • **name** [string] The name identifying the credential
> > > • **type** [string] The credential's type.
> > > • **username** [string] The username for the credential.
> > > • **description** [string] A long description of the credential.
> > > • **owner** [string] The username of the user who this credential belongs to. Using user.username is preferred.
> > > • **user** [dict::]
> > >
> > > > – **id** [integer] The ID of this user.
> > > >
> > > > – **name** [string] This user's name.
> > > >
> > > > – **username** [string] This user's username.
> > > >
> > > > – **initials** [string] This user's initials.
> > > >
> > > > – **online** [boolean] Whether this user is online.

- **remote_host_id** [integer] The ID of the remote host associated with this credential.
- **remote_host_name** [string] The name of the remote host associated with this credential.
- **state** [string] The U.S. state for the credential. Only for VAN credentials.
- **created_at** [string/time] The creation time for this credential.
- **updated_at** [string/time] The last modification time for this credential.
- **default** [boolean] Whether or not the credential is a default. Only for Database credentials.

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

Set the permissions groups has on this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>> **group_ids** [list] An array of one or more group IDs.
>> **permission_level** [string] Options are: "read", "write", or "manage".
>> **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
>> **send_shared_email** [boolean, optional] Send email to the recipients of a share.
>
> **Returns**
> [*civis.response.Response*](#)
>> - **readers** [dict::]
>>> – **users** [list::]
>>>> * id : integer
>>>> * name : string
>>> – **groups** [list::]
>>>> * id : integer
>>>> * name : string
>> - **writers** [dict::]
>>> – **users** [list::]
>>>> * id : integer
>>>> * name : string
>>> – **groups** [list::]
>>>> * id : integer
>>>> * name : string
>> - **owners** [dict::]
>>> – **users** [list::]
>>>> * id : integer
>>>> * name : string
>>> – **groups** [list::]
>>>> * id : integer
>>>> * name : string
>> - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

> • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

Set the permissions users have on this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>> **user_ids** [list] An array of one or more user IDs.
>> **permission_level** [string] Options are: "read", "write", or "manage".
>> **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
>> **send_shared_email** [boolean, optional] Send email to the recipients of a share.
>
> **Returns**
>
> [*civis.response.Response*](#)
>> • **readers** [dict::]
>>> – **users** [list::]
>>>> * id : integer
>>>> * name : string
>>> – **groups** [list::]
>>>> * id : integer
>>>> * name : string
>> • **writers** [dict::]
>>> – **users** [list::]
>>>> * id : integer
>>>> * name : string
>>> – **groups** [list::]
>>>> * id : integer
>>>> * name : string
>> • **owners** [dict::]
>>> – **users** [list::]
>>>> * id : integer
>>>> * name : string
>>> – **groups** [list::]
>>>> * id : integer
>>>> * name : string
>> • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
>> • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_transfer(*id*, *user_id*, *include_dependencies*, *, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

Transfer ownership of this object to another user

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>> **user_id** [integer] ID of target user

> **include_dependencies** [boolean] Whether or not to give manage permissions on all dependencies
>
> **email_body** [string, optional] Custom body text for e-mail sent on transfer.
>
> **send_email** [boolean, optional] Send email to the target user of the transfer?

> **Returns**
>
> *civis.response.Response*
>
> > • **dependencies** [list::] Dependent objects for this object - object_type : string
> >
> > > Dependent object type
> >
> > > – **fco_type** [string] Human readable dependent object type
> > >
> > > – **id** [integer] Dependent object ID
> > >
> > > – **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > >
> > > – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > >
> > > – **shared** [boolean] Whether dependent object was successfully shared with target user

Databases

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

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.databases.list(...)
```

Methods

get(id)	Show database information
get_whitelist_ips(id, whitelisted_ip_id)	View details about a whitelisted IP
list()	List databases
list_advanced_settings(id)	Get the advanced settings for this database
list_schemas(id, *[, name, credential_id])	List schemas in this database
list_whitelist_ips(id)	List whitelisted IPs for the specified database
patch_advanced_settings(id, *[, ...])	Update the advanced settings for this database
post_schemas_scan(id, schema, *[, ...])	Creates and enqueues a schema scanner job
put_advanced_settings(id, ex-port_caching_enabled)	Edit the advanced settings for this database

——

get(*id*)

> Show database information
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the database.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **id** [integer] The ID for the database.
> > > > • **name** [string] The name of the database.
> > > > • **adapter** [string] The type of the database.

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**
> >
> > > *civis.response.Response*
> > >
> > > > • **id** [integer] The ID of this whitelisted IP address.
> > > > • **remote_host_id** [integer] The ID of the database this rule is applied to.
> > > > • **security_group_id** [string] The ID of the security group this rule is applied to.
> > > > • **subnet_mask** [string] The subnet mask that is allowed by this rule.
> > > > • **authorized_by** [string] The user who authorized this rule.
> > > > • **is_active** [boolean] True if the rule is applied, false if it has been revoked.
> > > > • **created_at** [string/time] The time this rule was created.
> > > > • **updated_at** [string/time] The time this rule was last updated.

list()

> List databases
>
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **id** [integer] The ID for the database.
> > > > • **name** [string] The name of the database.
> > > > • **adapter** [string] The type of the database.

list_advanced_settings(*id*)

> Get the advanced settings for this database
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the database this advanced settings object belongs to.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **export_caching_enabled** [boolean] Whether or not caching is enabled for export jobs run on this database server.

list_schemas(*id*, ***, *name='DEFAULT'*, *credential_id='DEFAULT'*)

> List schemas in this database
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the database.
> > >
> > > **name** [string, optional] If specified, will be used to filter the schemas returned. Substring matching is supported (e.g., "name=schema" will return both "schema1" and "schema2").
> > >
> > > **credential_id** [integer, optional] If provided, schemas will be filtered based on the given credential.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **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**
>>> *civis.response.Response*
>>>> - **id** [integer] The ID of this whitelisted IP address.
>>>> - **remote_host_id** [integer] The ID of the database this rule is applied to.
>>>> - **security_group_id** [string] The ID of the security group this rule is applied to.
>>>> - **subnet_mask** [string] The subnet mask that is allowed by this rule.
>>>> - **created_at** [string/time] The time this rule was created.
>>>> - **updated_at** [string/time] The time this rule was last updated.

patch_advanced_settings(*id*, ***, *export_caching_enabled='DEFAULT'*)

> Update the advanced settings for this database
>> **Parameters**
>>> **id** [integer] The ID of the database this advanced settings object belongs to.
>>>
>>> **export_caching_enabled** [boolean, optional] Whether or not caching is enabled for export jobs run on this database server.
>>
>> **Returns**
>>> *civis.response.Response*
>>>> - **export_caching_enabled** [boolean] Whether or not caching is enabled for export jobs run on this database server.

post_schemas_scan(*id*, *schema*, ***, *stats_priority='DEFAULT'*)

> Creates and enqueues a schema scanner job
>> **Parameters**
>>> **id** [integer] The ID of the database.
>>>
>>> **schema** [string] The name of the schema.
>>>
>>> **stats_priority** [string, optional] When to sync table statistics for every table in the schema. Valid options are the following. Option: 'flag' means to flag stats for the next scheduled run of a full table scan on the database. Option: 'block' means to block this job on stats syncing. Option: 'queue' means to queue a separate job for syncing stats and do not block this job on the queued job. Defaults to 'flag'
>>
>> **Returns**
>>> *civis.response.Response*
>>>> - **job_id** [integer] The ID of the job created.
>>>> - **run_id** [integer] The ID of the run created.

put_advanced_settings(*id*, *export_caching_enabled*)

> Edit the advanced settings for this database
>> **Parameters**
>>> **id** [integer] The ID of the database this advanced settings object belongs to.
>>>
>>> **export_caching_enabled** [boolean] Whether or not caching is enabled for export jobs run on this database server.
>>
>> **Returns**
>>> *civis.response.Response*
>>>> - **export_caching_enabled** [boolean] Whether or not caching is enabled for export jobs run on this database server.

Endpoints

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

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.endpoints.list(...)
```

Methods

list()	List API endpoints

—

list()

> List API endpoints
>
> > **Returns**
> >
> > > **None** Response code 200: success

Enhancements

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

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.enhancements.post_civis_data_match(...)
```

Methods

delete_cass_ncoa_projects(id, project_id)	Remove a CASS/NCOA Enhancement from a project
delete_cass_ncoa_runs(id, run_id)	Cancel a run
delete_cass_ncoa_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_cass_ncoa_shares_users(id, user_id)	Revoke the permissions a user has on this object
delete_civis_data_match_projects(id, project_id)	Remove a Civis Data Match Enhancement from a project
delete_civis_data_match_runs(id, run_id)	Cancel a run
delete_civis_data_match_shares_groups(id, ...)	Revoke the permissions a group has on this object

delete_civis_data_match_shares_users(id, user_id)	Revoke the permissions a user has on this object
delete_geocode_projects(id, project_id)	Remove a Geocode Enhancement from a project
delete_geocode_runs(id, run_id)	Cancel a run
delete_geocode_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_geocode_shares_users(id, user_id)	Revoke the permissions a user has on this object
get_cass_ncoa(id)	Get a CASS/NCOA Enhancement
get_cass_ncoa_runs(id, run_id)	Check status of a run
get_civis_data_match(id)	Get a Civis Data Match Enhancement
get_civis_data_match_runs(id, run_id)	Check status of a run
get_geocode(id)	Get a Geocode Enhancement
get_geocode_runs(id, run_id)	Check status of a run
list(*[, type, author, status, archived, ...])	List Enhancements
list_cass_ncoa_dependencies(id, *[, user_id])	List dependent objects for this object
list_cass_ncoa_projects(id, *[, hidden])	List the projects a CASS/NCOA Enhancement belongs to
list_cass_ncoa_runs(id, *[, limit, ...])	List runs for the given cass_ncoa
list_cass_ncoa_runs_logs(id, run_id, *[, ...])	Get the logs for a run
list_cass_ncoa_runs_outputs(id, run_id, *[, ...])	List the outputs for a run
list_cass_ncoa_shares(id)	List users and groups permissioned on this object
list_civis_data_match_dependencies(id, *[, ...])	List dependent objects for this object
list_civis_data_match_projects(id, *[, hidden])	List the projects a Civis Data Match Enhancement belongs to
list_civis_data_match_runs(id, *[, limit, ...])	List runs for the given civis_data_match
list_civis_data_match_runs_logs(id, run_id, *)	Get the logs for a run
list_civis_data_match_runs_outputs(id, run_id, *)	List the outputs for a run
list_civis_data_match_shares(id)	List users and groups permissioned on this object
list_field_mapping()	List the fields in a field mapping for Civis Data Match, Data Unification, and Table Deduplication jobs
list_geocode_dependencies(id, *[, user_id])	List dependent objects for this object
list_geocode_projects(id, *[, hidden])	List the projects a Geocode Enhancement belongs to
list_geocode_runs(id, *[, limit, page_num, ...])	List runs for the given geocode
list_geocode_runs_logs(id, run_id, *[, ...])	Get the logs for a run
list_geocode_runs_outputs(id, run_id, *[, ...])	List the outputs for a run
list_geocode_shares(id)	List users and groups permissioned on this object
list_types()	List available enhancement types
patch_cass_ncoa(id, *[, name, schedule, ...])	Update some attributes of this CASS/NCOA Enhancement
patch_civis_data_match(id, *[, name, ...])	Update some attributes of this Civis Data Match Enhancement
patch_geocode(id, *[, name, schedule, ...])	Update some attributes of this Geocode Enhancement
post_cass_ncoa(name, source, *[, schedule, ...])	Create a CASS/NCOA Enhancement
post_cass_ncoa_cancel(id)	Cancel a run
post_cass_ncoa_runs(id)	Start a run
post_civis_data_match(name, ...[, schedule, ...])	Create a Civis Data Match Enhancement

<div align="center">Table 21 – continued from previous page</div>

post_civis_data_match_cancel(id)	Cancel a run
post_civis_data_match_clone(id, *[, ...])	Clone this Civis Data Match Enhancement
post_civis_data_match_runs(id)	Start a run
post_geocode(name, remote_host_id, ...[, ...])	Create a Geocode Enhancement
post_geocode_cancel(id)	Cancel a run
post_geocode_runs(id)	Start a run
put_cass_ncoa(id, name, source, *[, ...])	Replace all attributes of this CASS/NCOA Enhancement
put_cass_ncoa_archive(id, status)	Update the archive status of this object
put_cass_ncoa_projects(id, project_id)	Add a CASS/NCOA Enhancement to a project
put_cass_ncoa_shares_groups(id, group_ids, ...)	Set the permissions groups has on this object
put_cass_ncoa_shares_users(id, user_ids, ...)	Set the permissions users have on this object
put_cass_ncoa_transfer(id, user_id, ...[, ...])	Transfer ownership of this object to another user
put_civis_data_match(id, name, ...[, ...])	Replace all attributes of this Civis Data Match Enhancement
put_civis_data_match_archive(id, status)	Update the archive status of this object
put_civis_data_match_projects(id, project_id)	Add a Civis Data Match Enhancement to a project
put_civis_data_match_shares_groups(id, ...)	Set the permissions groups has on this object
put_civis_data_match_shares_users(id, ...[, ...])	Set the permissions users have on this object
put_civis_data_match_transfer(id, user_id, ...)	Transfer ownership of this object to another user
put_geocode(id, name, remote_host_id, ...[, ...])	Replace all attributes of this Geocode Enhancement
put_geocode_archive(id, status)	Update the archive status of this object
put_geocode_projects(id, project_id)	Add a Geocode Enhancement to a project
put_geocode_shares_groups(id, group_ids, ...)	Set the permissions groups has on this object
put_geocode_shares_users(id, user_ids, ...)	Set the permissions users have on this object
put_geocode_transfer(id, user_id, ...[, ...])	Transfer ownership of this object to another user

——

delete_cass_ncoa_projects(*id*, *project_id*)

Remove a CASS/NCOA Enhancement from a project

> **Parameters**
>> **id** [integer] The ID of the CASS/NCOA Enhancement.
>> **project_id** [integer] The ID of the project.
> **Returns**
>> **None** Response code 204: success

delete_cass_ncoa_runs(*id*, *run_id*)

Cancel a run

> **Parameters**
>> **id** [integer] The ID of the cass_ncoa.
>> **run_id** [integer] The ID of the run.
> **Returns**
>> **None** Response code 202: success

delete_cass_ncoa_shares_groups(*id*, *group_id*)

Revoke the permissions a group has on this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.

group_id [integer] The ID of the group.

> **Returns**
> > **None** Response code 204: success

delete_cass_ncoa_shares_users(*id*, *user_id*)

> Revoke the permissions a user has on this object
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_id** [integer] The ID of the user.
> >
> > **Returns**
> > > **None** Response code 204: success

delete_civis_data_match_projects(*id*, *project_id*)

> Remove a Civis Data Match Enhancement from a project
> > **Parameters**
> > > **id** [integer] The ID of the Civis Data Match Enhancement.
> > > **project_id** [integer] The ID of the project.
> >
> > **Returns**
> > > **None** Response code 204: success

delete_civis_data_match_runs(*id*, *run_id*)

> Cancel a run
> > **Parameters**
> > > **id** [integer] The ID of the civis_data_match.
> > > **run_id** [integer] The ID of the run.
> >
> > **Returns**
> > > **None** Response code 202: success

delete_civis_data_match_shares_groups(*id*, *group_id*)

> Revoke the permissions a group has on this object
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **group_id** [integer] The ID of the group.
> >
> > **Returns**
> > > **None** Response code 204: success

delete_civis_data_match_shares_users(*id*, *user_id*)

> Revoke the permissions a user has on this object
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_id** [integer] The ID of the user.
> >
> > **Returns**
> > > **None** Response code 204: success

delete_geocode_projects(*id*, *project_id*)

> Remove a Geocode Enhancement from a project
> > **Parameters**
> > > **id** [integer] The ID of the Geocode Enhancement.
> > > **project_id** [integer] The ID of the project.
> >
> > **Returns**
> > > **None** Response code 204: success

delete_geocode_runs(*id*, *run_id*)

> Cancel a run
> > **Parameters**
> > > **id** [integer] The ID of the geocode.
> > > **run_id** [integer] The ID of the run.

> **Returns**
> > **None** Response code 202: success

delete_geocode_shares_groups(*id*, *group_id*)

> Revoke the permissions a group has on this object
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **group_id** [integer] The ID of the group.
> > **Returns**
> > > **None** Response code 204: success

delete_geocode_shares_users(*id*, *user_id*)

> Revoke the permissions a user has on this object
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_id** [integer] The ID of the user.
> > **Returns**
> > > **None** Response code 204: success

get_cass_ncoa(*id*)

> Get a CASS/NCOA Enhancement
> > **Parameters**
> > > **id** [integer]
> > **Returns**
> > > [*civis.response.Response*](#)
> > > > - **id** [integer] The ID for the enhancement.
> > > > - **name** [string] The name of the enhancement job.
> > > > - **type** [string] The type of the enhancement (e.g CASS-NCOA)
> > > > - **created_at** [string/time] The time this enhancement was created.
> > > > - **updated_at** [string/time] The time the enhancement was last updated.
> > > > - **author** [dict::]
> > > >
> > > > > - **id** [integer] The ID of this user.
> > > > >
> > > > > - **name** [string] This user's name.
> > > > >
> > > > > - **username** [string] This user's username.
> > > > >
> > > > > - **initials** [string] This user's initials.
> > > > >
> > > > > - **online** [boolean] Whether this user is online.
> > > > - **state** [string] The status of the enhancement's last run
> > > > - **schedule** [dict::]
> > > >
> > > > > - **scheduled** [boolean] If the item is scheduled.
> > > > >
> > > > > - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > > > >
> > > > > - **scheduled_hours** [list] Hours of the day it is scheduled on.
> > > > >
> > > > > - **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> > > > >
> > > > > - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
> > > > >
> > > > > - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
> > > > - **parent_id** [integer] Parent ID that triggers this enhancement.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **source** [dict::]

    - **database_table** [dict::]

        * **schema** [string] The schema name of the source table.

        * **table** [string] The name of the source table.

        * **remote_host_id** [integer] The ID of the database host for the table.

        * **credential_id** [integer] The id of the credentials to be used when performing the enhancement.

        * **multipart_key** [list] The source table primary key.
- **destination** [dict::]

    - **database_table** [dict::]

        * **schema** [string] The schema name for the output data.

        * **table** [string] The table name for the output data.
- **column_mapping** [dict::]

    - **address1** [string] The first address line.

- **address2** [string] The second address line.

- **city** [string] The city of an address.

- **state** [string] The state of an address.

- **zip** [string] The zip code of an address.

- **name** [string] The full name of the resident at this address. If needed, separate multiple columns with +, e.g. *first_name+last_name*

- **company** [string] The name of the company located at this address.

- **use_default_column_mapping** [boolean] Defaults to true, where the existing column mapping on the input table will be used. If false, a custom column mapping must be provided.

- **perform_ncoa** [boolean] Whether to update addresses for records matching the National Change of Address (NCOA) database.

- **ncoa_credential_id** [integer] Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

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

- **limiting_sql** [string] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').

- **archived** [string] The archival status of the requested item(s).

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

get_cass_ncoa_runs(*id*, *run_id*)

Check status of a run

Parameters

id [integer] The ID of the cass_ncoa.

run_id [integer] The ID of the run.

Returns

civis.response.Response

- **id** [integer] The ID of the run.
- **cass_ncoa_id** [integer] The ID of the cass_ncoa.
- **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **is_cancel_requested** [boolean] True if run cancel requested, else false.
- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.
- **error** [string] The error, if any, returned by the run.

get_civis_data_match(*id*)

Get a Civis Data Match Enhancement

Parameters

id [integer]

Returns

civis.response.Response

- **id** [integer] The ID for the enhancement.
- **name** [string] The name of the enhancement job.
- **type** [string] The type of the enhancement (e.g CASS-NCOA)
- **created_at** [string/time] The time this enhancement was created.

- **updated_at** [string/time] The time the enhancement was last updated.
- **author** [dict::]
    - **id** [integer] The ID of this user.
    - **name** [string] This user's name.
    - **username** [string] This user's username.
    - **initials** [string] This user's initials.
    - **online** [boolean] Whether this user is online.
- **state** [string] The status of the enhancement's last run
- **schedule** [dict::]
    - **scheduled** [boolean] If the item is scheduled.
    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
    - **scheduled_hours** [list] Hours of the day it is scheduled on.
    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.
    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **parent_id** [integer] Parent ID that triggers this enhancement.
- **notifications** [dict::]
    - **urls** [list] URLs to receive a POST request at job completion
    - **success_email_subject** [string] Custom subject line for success e-mail.
    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.
    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.
    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."
    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.
    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.
    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.
    - **success_on** [boolean] If success email notifications are on.
    - **failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]
    - **id** [integer] The ID of this user.
    - **name** [string] This user's name.

> > > > > > > > – **username** [string] This user's username.
> > > > > > > >
> > > > > > > > – **initials** [string] This user's initials.
> > > > > > > >
> > > > > > > > – **online** [boolean] Whether this user is online.
> > > > > > > - **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
> > > > > > > - **input_field_mapping** [dict] The field (i.e., column) mapping for the input table. See https://api.civisanalytics.com/enhancements/field-mapping for a list of valid field types and descriptions. Each field type should be mapped to a string specifying a column name in the input table. For field types that support multiple values (e.g., the "phone" field), a list of column names can be provided (e.g., {"phone": ["home_phone", "mobile_phone"], … }).
> > > > > > > - **input_table** [dict::]
> > > > > > > > – **database_name** [string] The Redshift database name for the table.
> > > > > > > >
> > > > > > > > – **schema** [string] The schema name for the table.
> > > > > > > >
> > > > > > > > – **table** [string] The table name.
> > > > > > > - **match_target_id** [integer] The ID of the Civis Data match target. See /match_targets for IDs.
> > > > > > > - **output_table** [dict::]
> > > > > > > > – **database_name** [string] The Redshift database name for the table.
> > > > > > > >
> > > > > > > > – **schema** [string] The schema name for the table.
> > > > > > > >
> > > > > > > > – **table** [string] The table name.
> > > > > > > - **max_matches** [integer] The maximum number of matches per record in the input table to return. Must be between 0 and 10. 0 returns all matches.
> > > > > > > - **threshold** [number/float] The score threshold (between 0 and 1). Matches below this threshold will not be returned. The default value is 0.5.
> > > > > > > - **archived** [boolean] Whether the Civis Data Match Job has been archived.
> > > > > > > - **last_run** [dict::]
> > > > > > > > – id : integer
> > > > > > > >
> > > > > > > > – state : string
> > > > > > > >
> > > > > > > > – **created_at** [string/time] The time that the run was queued.
> > > > > > > >
> > > > > > > > – **started_at** [string/time] The time that the run started.
> > > > > > > >
> > > > > > > > – **finished_at** [string/time] The time that the run completed.
> > > > > > > >
> > > > > > > > – **error** [string] The error message for this run, if present.

get_civis_data_match_runs(*id*, *run_id*)

  Check status of a run

   Parameters

   id [integer] The ID of the civis_data_match.

   run_id [integer] The ID of the run.

   Returns

   civis.response.Response

   - **id** [integer] The ID of the run.
   - **civis_data_match_id** [integer] The ID of the civis_data_match.

- **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **is_cancel_requested** [boolean] True if run cancel requested, else false.
- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.
- **error** [string] The error, if any, returned by the run.

get_geocode(*id*)

Get a Geocode Enhancement

    Parameters

        id [integer]

    Returns

        civis.response.Response

- **id** [integer] The ID for the enhancement.
- **name** [string] The name of the enhancement job.
- **type** [string] The type of the enhancement (e.g CASS-NCOA)
- **created_at** [string/time] The time this enhancement was created.
- **updated_at** [string/time] The time the enhancement was last updated.
- **author** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.
- **state** [string] The status of the enhancement's last run
- **schedule** [dict::]

  – **scheduled** [boolean] If the item is scheduled.

  – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  – **scheduled_hours** [list] Hours of the day it is scheduled on.

  – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

  – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

  – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **parent_id** [integer] Parent ID that triggers this enhancement.
- **notifications** [dict::]

  – **urls** [list] URLs to receive a POST request at job completion

  – **success_email_subject** [string] Custom subject line for success e-mail.

  – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

- **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

- **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **remote_host_id** [integer] The ID of the remote host.
- **credential_id** [integer] The ID of the remote host credential.
- **source_schema_and_table** [string] The source database schema and table.
- **multipart_key** [list] The source table primary key.
- **limiting_sql** [string] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').
- **target_schema** [string] The output table schema.
- **target_table** [string] The output table name.
- **country** [string] The country of the addresses to be geocoded; either 'us' or 'ca'.
- **provider** [string] The geocoding provider; one of postgis, nominatim, and geocoder_ca.
- **output_address** [boolean] Whether to output the parsed address. Only guaranteed for the 'postgis' provider.
- **archived** [string] The archival status of the requested item(s).
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

get_geocode_runs(*id*, *run_id*)

Check status of a run

Parameters

id [integer] The ID of the geocode.

run_id [integer] The ID of the run.

Returns

civis.response.Response

- **id** [integer] The ID of the run.
- **geocode_id** [integer] The ID of the geocode.
- **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **is_cancel_requested** [boolean] True if run cancel requested, else false.
- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.

- **error** [string] The error, if any, returned by the run.

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

> List Enhancements
>
> > **Parameters**
> >
> > > **type** [string, optional] If specified, return items of these types.
> > >
> > > **author** [string, optional] If specified, return items from any of these authors. It accepts a comma- separated list of user IDs.
> > >
> > > **status** [string, optional] If specified, returns items 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 item(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**
> >
> > *civis.response.PaginatedResponse*
> >
> > > - **id** [integer] The ID for the enhancement.
> > > - **name** [string] The name of the enhancement job.
> > > - **type** [string] The type of the enhancement (e.g CASS-NCOA)
> > > - **created_at** [string/time] The time this enhancement was created.
> > > - **updated_at** [string/time] The time the enhancement was last updated.
> > > - **author** [dict::]
> > >
> > > > - **id** [integer] The ID of this user.
> > > >
> > > > - **name** [string] This user's name.
> > > >
> > > > - **username** [string] This user's username.
> > > >
> > > > - **initials** [string] This user's initials.
> > > >
> > > > - **online** [boolean] Whether this user is online.
> > >
> > > - **state** [string] The status of the enhancement's last run
> > > - **archived** [string] The archival status of the requested item(s).

list_cass_ncoa_dependencies(*id*, *, *user_id='DEFAULT'*)

> List dependent objects for this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **user_id** [integer, optional] ID of target user
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > - **object_type** [string] Dependent object type
> > > - **fco_type** [string] Human readable dependent object type
> > > - **id** [integer] Dependent object ID
> > > - **name** [string] Dependent object name, or nil if the requesting user cannot read this object

> - **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> - **shareable** [boolean] Whether or not the requesting user can share this object.

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

> List the projects a CASS/NCOA Enhancement belongs to
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the CASS/NCOA Enhancement.
> > >
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > - **id** [integer] The ID for this project.
> > > > - **author** [dict::]
> > > >
> > > > > - **id** [integer] The ID of this user.
> > > > >
> > > > > - **name** [string] This user's name.
> > > > >
> > > > > - **username** [string] This user's username.
> > > > >
> > > > > - **initials** [string] This user's initials.
> > > > >
> > > > > - **online** [boolean] Whether this user is online.
> > > >
> > > > - **name** [string] The name of this project.
> > > > - **description** [string] A description of the project.
> > > > - **users** [list::] Users who can see the project. - id : integer
> > > >
> > > > > The ID of this user.
> > > >
> > > > > - **name** [string] This user's name.
> > > > >
> > > > > - **username** [string] This user's username.
> > > > >
> > > > > - **initials** [string] This user's initials.
> > > > >
> > > > > - **online** [boolean] Whether this user is online.
> > > >
> > > > - auto_share : boolean
> > > > - created_at : string/time
> > > > - updated_at : string/time
> > > > - **archived** [string] The archival status of the requested item(s).

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

> List runs for the given cass_ncoa
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the cass_ncoa.
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 100.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to id. Must be one of: id.
> > >
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > >
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

> **Returns**
> > `civis.response.PaginatedResponse`
> > > - **id** [integer] The ID of the run.
> > > - **cass_ncoa_id** [integer] The ID of the cass_ncoa.
> > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > - **started_at** [string/time] The time the last run started at.
> > > - **finished_at** [string/time] The time the last run completed.
> > > - **error** [string] The error, if any, returned by the run.

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

> Get the logs for a run
> > **Parameters**
> > > **id** [integer] The ID of the cass_ncoa.
> > > **run_id** [integer] The ID of the run.
> > > **last_id** [integer, optional] The ID of the last log message received. Log entries with this ID value or lower will be omitted.Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.
> > > **limit** [integer, optional] The maximum number of log messages to return. Default of 10000.
> > **Returns**
> > > `civis.response.Response`
> > > > - **id** [integer] The ID of the log.
> > > > - **created_at** [string/date-time] The time the log was created.
> > > > - **message** [string] The log message.
> > > > - **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

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

> List the outputs for a run
> > **Parameters**
> > > **id** [integer] The ID of the job.
> > > **run_id** [integer] The ID of the run.
> > > **limit** [integer, optional] Number of results to return. Defaults to its maximum of 50.
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > > **order** [string, optional] The field on which to order the result set. Defaults to created_at. Must be one of: created_at, id.
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> > **Returns**
> > > `civis.response.PaginatedResponse`
> > > > - **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
> > > > - **object_id** [integer] The ID of the output.
> > > > - **name** [string] The name of the output.
> > > > - **link** [string] The hypermedia link to the output.
> > > > - value : string

list_cass_ncoa_shares(*id*)

> List users and groups permissioned on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > • **readers** [dict::]
> > >
> > > > – **users** [list::]
> > > >
> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string
> > > >
> > > > – **groups** [list::]
> > > >
> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string
> > >
> > > • **writers** [dict::]
> > >
> > > > – **users** [list::]
> > > >
> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string
> > > >
> > > > – **groups** [list::]
> > > >
> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string
> > >
> > > • **owners** [dict::]
> > >
> > > > – **users** [list::]
> > > >
> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string
> > > >
> > > > – **groups** [list::]
> > > >
> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string
> > >
> > > • **total_user_shares** [integer] For owners, the number of total users shared.
> > >   For writers and readers, the number of visible users shared.
> > > • **total_group_shares** [integer] For owners, the number of total groups
> > >   shared. For writers and readers, the number of visible groups shared.

list_civis_data_match_dependencies(*id*, ***, *user_id='DEFAULT'*)

> List dependent objects for this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **user_id** [integer, optional] ID of target user
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > • **object_type** [string] Dependent object type
> > > • **fco_type** [string] Human readable dependent object type
> > > • **id** [integer] Dependent object ID
> > > • **name** [string] Dependent object name, or nil if the requesting user cannot
> > >   read this object

- **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
- **shareable** [boolean] Whether or not the requesting user can share this object.

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

List the projects a Civis Data Match Enhancement belongs to

> **Parameters**
>> **id** [integer] The ID of the Civis Data Match Enhancement.
>> **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
>
> **Returns**
>> [*civis.response.Response*](#)
>>> - **id** [integer] The ID for this project.
>>> - **author** [dict::]
>>>
>>>> – **id** [integer] The ID of this user.
>>>>
>>>> – **name** [string] This user's name.
>>>>
>>>> – **username** [string] This user's username.
>>>>
>>>> – **initials** [string] This user's initials.
>>>>
>>>> – **online** [boolean] Whether this user is online.
>>>
>>> - **name** [string] The name of this project.
>>> - **description** [string] A description of the project.
>>> - **users** [list::] Users who can see the project. - id : integer
>>>
>>>> The ID of this user.
>>>
>>>> – **name** [string] This user's name.
>>>>
>>>> – **username** [string] This user's username.
>>>>
>>>> – **initials** [string] This user's initials.
>>>>
>>>> – **online** [boolean] Whether this user is online.
>>>
>>> - auto_share : boolean
>>> - created_at : string/time
>>> - updated_at : string/time
>>> - **archived** [string] The archival status of the requested item(s).

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

List runs for the given civis_data_match

> **Parameters**
>> **id** [integer] The ID of the civis_data_match.
>> **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**
> > *civis.response.PaginatedResponse*
> > > - **id** [integer] The ID of the run.
> > > - **civis_data_match_id** [integer] The ID of the civis_data_match.
> > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > - **started_at** [string/time] The time the last run started at.
> > > - **finished_at** [string/time] The time the last run completed.
> > > - **error** [string] The error, if any, returned by the run.

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

> Get the logs for a run
> > **Parameters**
> > > **id** [integer] The ID of the civis_data_match.
> > > **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**
> > > *civis.response.Response*
> > > > - **id** [integer] The ID of the log.
> > > > - **created_at** [string/date-time] The time the log was created.
> > > > - **message** [string] The log message.
> > > > - **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

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

> List the outputs for a run
> > **Parameters**
> > > **id** [integer] The ID of the job.
> > > **run_id** [integer] The ID of the run.
> > > **limit** [integer, optional] Number of results to return. Defaults to its maximum of 50.
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > > **order** [string, optional] The field on which to order the result set. Defaults to created_at. Must be one of: created_at, id.
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> >
> > **Returns**
> > > *civis.response.PaginatedResponse*
> > > > - **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
> > > > - **object_id** [integer] The ID of the output.
> > > > - **name** [string] The name of the output.
> > > > - **link** [string] The hypermedia link to the output.
> > > > - value : string

list_civis_data_match_shares(*id*)

> List users and groups permissioned on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **readers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **writers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **owners** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_field_mapping()

> List the fields in a field mapping for Civis Data Match, Data Unification, and Table Deduplication jobs
>
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **field** [string] The name of the field.
> > > > • **description** [string] The description of the field.

list_geocode_dependencies(*id*, ***, *user_id='DEFAULT'*)

> List dependent objects for this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_id** [integer, optional] ID of target user
> >
> > **Returns**
> >
> > > *civis.response.Response*

- **object_type** [string] Dependent object type
- **fco_type** [string] Human readable dependent object type
- **id** [integer] Dependent object ID
- **name** [string] Dependent object name, or nil if the requesting user cannot read this object
- **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
- **shareable** [boolean] Whether or not the requesting user can share this object.

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

> List the projects a Geocode Enhancement belongs to
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the Geocode Enhancement.
> > >
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > - **id** [integer] The ID for this project.
> > > > - **author** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > - **name** [string] The name of this project.
> > > > - **description** [string] A description of the project.
> > > > - **users** [list::] Users who can see the project. - id : integer
> > > >
> > > > > The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > - auto_share : boolean
> > > > - created_at : string/time
> > > > - updated_at : string/time
> > > > - **archived** [string] The archival status of the requested item(s).

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

> List runs for the given geocode
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the geocode.
> > >
> > > **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**
> >
> > *civis.response.PaginatedResponse*
> >
> > > - **id** [integer] The ID of the run.
> > > - **geocode_id** [integer] The ID of the geocode.
> > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > - **started_at** [string/time] The time the last run started at.
> > > - **finished_at** [string/time] The time the last run completed.
> > > - **error** [string] The error, if any, returned by the run.

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

> Get the logs for a run
>
> > **Parameters**
> >
> > **id** [integer] The ID of the geocode.
> >
> > **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**
> >
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID of the log.
> > > - **created_at** [string/date-time] The time the log was created.
> > > - **message** [string] The log message.
> > > - **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

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

> List the outputs for a run
>
> > **Parameters**
> >
> > **id** [integer] The ID of the job.
> >
> > **run_id** [integer] The ID of the run.
> >
> > **limit** [integer, optional] Number of results to return. Defaults to its maximum of 50.
> >
> > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> >
> > **order** [string, optional] The field on which to order the result set. Defaults to created_at. Must be one of: created_at, id.
> >
> > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> >
> > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> >
> > **Returns**
> >
> > *civis.response.PaginatedResponse*
> >
> > > - **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
> > > - **object_id** [integer] The ID of the output.

- **name** [string] The name of the output.
- **link** [string] The hypermedia link to the output.
- value : string

list_geocode_shares(*id*)

List users and groups permissioned on this object

    Parameters

        id [integer] The ID of the resource that is shared.

    Returns

        civis.response.Response

- **readers** [dict::]

  – **users** [list::]

    ∗ id : integer

    ∗ name : string

  – **groups** [list::]

    ∗ id : integer

    ∗ name : string

- **writers** [dict::]

  – **users** [list::]

    ∗ id : integer

    ∗ name : string

  – **groups** [list::]

    ∗ id : integer

    ∗ name : string

- **owners** [dict::]

  – **users** [list::]

    ∗ id : integer

    ∗ name : string

  – **groups** [list::]

    ∗ id : integer

    ∗ name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_types()

List available enhancement types

    Returns

        civis.response.Response

- **name** [string] The name of the type.

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

> Update some attributes of this CASS/NCOA Enhancement
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the enhancement.
> > >
> > > **name** [string, optional] The name of the enhancement job.
> > >
> > > **schedule** [dict, optional::]
> > >
> > > > • **scheduled** [boolean] If the item is scheduled.
> > > > • **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > > > • **scheduled_hours** [list] Hours of the day it is scheduled on.
> > > > • **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> > > > • **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
> > > > • **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
> > >
> > > **parent_id** [integer, optional] Parent ID that triggers this enhancement.
> > >
> > > **notifications** [dict, optional::]
> > >
> > > > • **urls** [list] URLs to receive a POST request at job completion
> > > > • **success_email_subject** [string] Custom subject line for success e-mail.
> > > > • **success_email_body** [string] Custom body text for success e-mail, written in Markdown.
> > > > • **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.
> > > > • **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."
> > > > • **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.
> > > > • **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.
> > > > • **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.
> > > > • **success_on** [boolean] If success email notifications are on.
> > > > • **failure_on** [boolean] If failure email notifications are on.
> > >
> > > **source** [dict, optional::]
> > >
> > > > • **database_table** [dict::]
> > > >
> > > > > – **schema** [string] The schema name of the source table.
> > > > >
> > > > > – **table** [string] The name of the source table.
> > > > >
> > > > > – **remote_host_id** [integer] The ID of the database host for the table.
> > > > >
> > > > > – **credential_id** [integer] The id of the credentials to be used when performing the enhancement.
> > > > >
> > > > > – **multipart_key** [list] The source table primary key.
> > >
> > > **destination** [dict, optional::]
> > >
> > > > • **database_table** [dict::]
> > > >
> > > > > – **schema** [string] The schema name for the output data.
> > > > >
> > > > > – **table** [string] The table name for the output data.

     column_mapping [dict, optional::]

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

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

     perform_ncoa [boolean, optional] Whether to update addresses for records matching the National Change of Address (NCOA) database.

     ncoa_credential_id [integer, optional] Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

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

     limiting_sql [string, optional] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').

Returns

    [*civis.response.Response*](#)

- **id** [integer] The ID for the enhancement.
- **name** [string] The name of the enhancement job.
- **type** [string] The type of the enhancement (e.g CASS-NCOA)
- **created_at** [string/time] The time this enhancement was created.
- **updated_at** [string/time] The time the enhancement was last updated.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

- **state** [string] The status of the enhancement's last run
- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

    – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    – **scheduled_hours** [list] Hours of the day it is scheduled on.

    – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **parent_id** [integer] Parent ID that triggers this enhancement.
- **notifications** [dict::]

    – **urls** [list] URLs to receive a POST request at job completion

    – **success_email_subject** [string] Custom subject line for success e-mail.

    – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    – **success_on** [boolean] If success email notifications are on.

    – **failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- **source** [dict::]

    – **database_table** [dict::]

        * **schema** [string] The schema name of the source table.

        * **table** [string] The name of the source table.

        * **remote_host_id** [integer] The ID of the database host for the table.

        * **credential_id** [integer] The id of the credentials to be used when performing the enhancement.

        * **multipart_key** [list] The source table primary key.
- **destination** [dict::]

    – **database_table** [dict::]

        * **schema** [string] The schema name for the output data.

        * **table** [string] The table name for the output data.
- **column_mapping** [dict::]

> > > > – **address1** [string] The first address line.
> > > >
> > > > – **address2** [string] The second address line.
> > > >
> > > > – **city** [string] The city of an address.
> > > >
> > > > – **state** [string] The state of an address.
> > > >
> > > > – **zip** [string] The zip code of an address.
> > > >
> > > > – **name** [string] The full name of the resident at this address. If needed, separate multiple columns with +, e.g. *first_name+last_name*
> > > >
> > > > – **company** [string] The name of the company located at this address.
> > >
> > > • **use_default_column_mapping** [boolean] Defaults to true, where the existing column mapping on the input table will be used. If false, a custom column mapping must be provided.
> > >
> > > • **perform_ncoa** [boolean] Whether to update addresses for records matching the National Change of Address (NCOA) database.
> > >
> > > • **ncoa_credential_id** [integer] Credential to use when performing NCOA updates. Required if 'performNcoa' is true.
> > >
> > > • **output_level** [string] The set of fields persisted by a CASS or NCOA enhancement.For CASS enhancements, one of 'cass' or 'all.'For NCOA enhancements, one of 'cass', 'ncoa' , 'coalesced' or 'all'.By default, all fields will be returned.
> > >
> > > • **limiting_sql** [string] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').
> > >
> > > • **archived** [string] The archival status of the requested item(s).
> > >
> > > • **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

patch_civis_data_match(*id*, ***, *name='DEFAULT'*, *schedule='DEFAULT'*, *parent_id='DEFAULT'*, *notifications='DEFAULT'*, *input_field_mapping='DEFAULT'*, *input_table='DEFAULT'*, *match_target_id='DEFAULT'*, *output_table='DEFAULT'*, *max_matches='DEFAULT'*, *threshold='DEFAULT'*, *archived='DEFAULT'*)

> Update some attributes of this Civis Data Match Enhancement
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the enhancement.
> > >
> > > **name** [string, optional] The name of the enhancement job.
> > >
> > > **schedule** [dict, optional::]
> > >
> > > > • **scheduled** [boolean] If the item is scheduled.
> > > >
> > > > • **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > > >
> > > > • **scheduled_hours** [list] Hours of the day it is scheduled on.
> > > >
> > > > • **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> > > >
> > > > • **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
> > > >
> > > > • **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
> > >
> > > **parent_id** [integer, optional] Parent ID that triggers this enhancement.
> > >
> > > **notifications** [dict, optional::]
> > >
> > > > • **urls** [list] URLs to receive a POST request at job completion
> > > >
> > > > • **success_email_subject** [string] Custom subject line for success e-mail.

- **success_email_body** [string] Custom body text for success e-mail, written in Markdown.
- **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.
- **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."
- **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.
- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.
- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.
- **success_on** [boolean] If success email notifications are on.
- **failure_on** [boolean] If failure email notifications are on.

input_field_mapping [dict, optional] The field (i.e., column) mapping for the input table. See https://api.civisanalytics.com/enhancements/field-mapping for a list of valid field types and descriptions. Each field type should be mapped to a string specifying a column name in the input table. For field types that support multiple values (e.g., the "phone" field), a list of column names can be provided (e.g., {"phone": ["home_phone", "mobile_phone"], . . . }).

input_table [dict, optional::]
- **database_name** [string] The Redshift database name for the table.
- **schema** [string] The schema name for the table.
- **table** [string] The table name.

match_target_id [integer, optional] The ID of the Civis Data match target. See /match_targets for IDs.

output_table [dict, optional::]
- **database_name** [string] The Redshift database name for the table.
- **schema** [string] The schema name for the table.
- **table** [string] The table name.

max_matches [integer, optional] The maximum number of matches per record in the input table to return. Must be between 0 and 10. 0 returns all matches.

threshold [number/float, optional] The score threshold (between 0 and 1). Matches below this threshold will not be returned. The default value is 0.5.

archived [boolean, optional] Whether the Civis Data Match Job has been archived.

Returns

civis.response.Response
- **id** [integer] The ID for the enhancement.
- **name** [string] The name of the enhancement job.
- **type** [string] The type of the enhancement (e.g CASS-NCOA)
- **created_at** [string/time] The time this enhancement was created.
- **updated_at** [string/time] The time the enhancement was last updated.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **state** [string] The status of the enhancement's last run
- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

- **scheduled_hours** [list] Hours of the day it is scheduled on.

- **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **parent_id** [integer] Parent ID that triggers this enhancement.
- **notifications** [dict::]

- **urls** [list] URLs to receive a POST request at job completion

- **success_email_subject** [string] Custom subject line for success e-mail.

- **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

- **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

- **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

- **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]

- **id** [integer] The ID of this user.

- **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **input_field_mapping** [dict] The field (i.e., column) mapping for the input table. See https://api.civisanalytics.com/enhancements/ field-mapping for a list of valid field types and descriptions. Each field type should be mapped to a string specifying a column name in the input table. For field types that support multiple values (e.g., the "phone" field), a list of column names can be provided (e.g., {"phone": ["home_phone", "mobile_phone"], ... }).

- **input_table** [dict::]

    - **database_name** [string] The Redshift database name for the table.

    - **schema** [string] The schema name for the table.

    - **table** [string] The table name.

- **match_target_id** [integer] The ID of the Civis Data match target. See /match_targets for IDs.

- **output_table** [dict::]

    - **database_name** [string] The Redshift database name for the table.

    - **schema** [string] The schema name for the table.

    - **table** [string] The table name.

- **max_matches** [integer] The maximum number of matches per record in the input table to return. Must be between 0 and 10. 0 returns all matches.

- **threshold** [number/float] The score threshold (between 0 and 1). Matches below this threshold will not be returned. The default value is 0.5.

- **archived** [boolean] Whether the Civis Data Match Job has been archived.

- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.

patch_geocode(*id*, *, *name='DEFAULT'*, *schedule='DEFAULT'*, *parent_id='DEFAULT'*,
    notifications='DEFAULT', *remote_host_id='DEFAULT'*, *credential_id='DEFAULT'*,
    source_schema_and_table='DEFAULT', *multipart_key='DEFAULT'*,
    limiting_sql='DEFAULT', *target_schema='DEFAULT'*, *target_table='DEFAULT'*,
    country='DEFAULT', *provider='DEFAULT'*, *output_address='DEFAULT'*)

    Update some attributes of this Geocode Enhancement

        Parameters

            id [integer] The ID for the enhancement.

            name [string, optional] The name of the enhancement job.

            schedule [dict, optional::]

- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

            parent_id [integer, optional] Parent ID that triggers this enhancement.

            notifications [dict, optional::]

- **urls** [list] URLs to receive a POST request at job completion

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

remote_host_id [integer, optional] The ID of the remote host.

credential_id [integer, optional] The ID of the remote host credential.

source_schema_and_table [string, optional] The source database schema and table.

multipart_key [list, optional] The source table primary key.

limiting_sql [string, optional] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').

target_schema [string, optional] The output table schema.

target_table [string, optional] The output table name.

country [string, optional] The country of the addresses to be geocoded; either 'us' or 'ca'.

provider [string, optional] The geocoding provider; one of postgis, nominatim, and geocoder_ca.

output_address [boolean, optional] Whether to output the parsed address. Only guaranteed for the 'postgis' provider.

Returns

 civis.response.Response

- **id** [integer] The ID for the enhancement.
- **name** [string] The name of the enhancement job.
- **type** [string] The type of the enhancement (e.g CASS-NCOA)
- **created_at** [string/time] The time this enhancement was created.
- **updated_at** [string/time] The time the enhancement was last updated.
- **author** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **state** [string] The status of the enhancement's last run
- **schedule** [dict::]

  - **scheduled** [boolean] If the item is scheduled.

  - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  - **scheduled_hours** [list] Hours of the day it is scheduled on.

  - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

  - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

  - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **parent_id** [integer] Parent ID that triggers this enhancement.
- **notifications** [dict::]

  - **urls** [list] URLs to receive a POST request at job completion

  - **success_email_subject** [string] Custom subject line for success e-mail.

  - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

  - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

  - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

  - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

  - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

  - **success_on** [boolean] If success email notifications are on.

  - **failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **remote_host_id** [integer] The ID of the remote host.
- **credential_id** [integer] The ID of the remote host credential.
- **source_schema_and_table** [string] The source database schema and table.
- **multipart_key** [list] The source table primary key.
- **limiting_sql** [string] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').
- **target_schema** [string] The output table schema.
- **target_table** [string] The output table name.
- **country** [string] The country of the addresses to be geocoded; either 'us' or 'ca'.
- **provider** [string] The geocoding provider; one of postgis, nominatim, and geocoder_ca.

> - **output_address** [boolean] Whether to output the parsed address. Only guaranteed for the 'postgis' provider.
> - **archived** [string] The archival status of the requested item(s).
> - **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

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

> Create a CASS/NCOA Enhancement
>> **Parameters**
>>> **name** [string] The name of the enhancement job.
>>> **source** [dict::]
>>>> - **database_table** [dict::]
>>>>
>>>>> – **schema** [string] The schema name of the source table.
>>>>>
>>>>> – **table** [string] The name of the source table.
>>>>>
>>>>> – **remote_host_id** [integer] The ID of the database host for the table.
>>>>>
>>>>> – **credential_id** [integer] The id of the credentials to be used when performing the enhancement.
>>>>>
>>>>> – **multipart_key** [list] The source table primary key.
>>>
>>> **schedule** [dict, optional::]
>>>> - **scheduled** [boolean] If the item is scheduled.
>>>> - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
>>>> - **scheduled_hours** [list] Hours of the day it is scheduled on.
>>>> - **scheduled_minutes** [list] Minutes of the day it is scheduled on.
>>>> - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
>>>> - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
>>>
>>> **parent_id** [integer, optional] Parent ID that triggers this enhancement.
>>> **notifications** [dict, optional::]
>>>> - **urls** [list] URLs to receive a POST request at job completion
>>>> - **success_email_subject** [string] Custom subject line for success e-mail.
>>>> - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.
>>>> - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.
>>>> - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."
>>>> - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.
>>>> - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.
>>>> - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.
>>>> - **success_on** [boolean] If success email notifications are on.
>>>> - **failure_on** [boolean] If failure email notifications are on.
>>>
>>> **destination** [dict, optional::]
>>>> - **database_table** [dict::]

> – **schema** [string] The schema name for the output data.
>
> – **table** [string] The table name for the output data.

column_mapping [dict, optional::]

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

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

perform_ncoa [boolean, optional] Whether to update addresses for records matching the National Change of Address (NCOA) database.

ncoa_credential_id [integer, optional] Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

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

limiting_sql [string, optional] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').

Returns

[*civis.response.Response*](#)

- **id** [integer] The ID for the enhancement.
- **name** [string] The name of the enhancement job.
- **type** [string] The type of the enhancement (e.g CASS-NCOA)
- **created_at** [string/time] The time this enhancement was created.
- **updated_at** [string/time] The time the enhancement was last updated.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

- **state** [string] The status of the enhancement's last run
- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

    – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    – **scheduled_hours** [list] Hours of the day it is scheduled on.

    – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

– **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **parent_id** [integer] Parent ID that triggers this enhancement.
- **notifications** [dict::]

    – **urls** [list] URLs to receive a POST request at job completion

    – **success_email_subject** [string] Custom subject line for success e-mail.

    – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    – **success_on** [boolean] If success email notifications are on.

    – **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

- **source** [dict::]

    – **database_table** [dict::]

        ∗ **schema** [string] The schema name of the source table.

        ∗ **table** [string] The name of the source table.

        ∗ **remote_host_id** [integer] The ID of the database host for the table.

        ∗ **credential_id** [integer] The id of the credentials to be used when performing the enhancement.

        ∗ **multipart_key** [list] The source table primary key.

- **destination** [dict::]

    – **database_table** [dict::]

        ∗ **schema** [string] The schema name for the output data.

        ∗ **table** [string] The table name for the output data.

- **column_mapping** [dict::]

    - **address1** [string] The first address line.

    - **address2** [string] The second address line.

    - **city** [string] The city of an address.

    - **state** [string] The state of an address.

    - **zip** [string] The zip code of an address.

    - **name** [string] The full name of the resident at this address. If needed, separate multiple columns with +, e.g. *first_name+last_name*

    - **company** [string] The name of the company located at this address.

- **use_default_column_mapping** [boolean] Defaults to true, where the existing column mapping on the input table will be used. If false, a custom column mapping must be provided.

- **perform_ncoa** [boolean] Whether to update addresses for records matching the National Change of Address (NCOA) database.

- **ncoa_credential_id** [integer] Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

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

- **limiting_sql** [string] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').

- **archived** [string] The archival status of the requested item(s).

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

post_cass_ncoa_cancel(*id*)

> Cancel a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the job.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > - **id** [integer] The ID of the run.
> > > - **state** [string] The state of the run, one of 'queued', 'running' or 'cancelled'.
> > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.

post_cass_ncoa_runs(*id*)

> Start a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the cass_ncoa.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > - **id** [integer] The ID of the run.
> > > - **cass_ncoa_id** [integer] The ID of the cass_ncoa.
> > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > - **started_at** [string/time] The time the last run started at.

> - **finished_at** [string/time] The time the last run completed.
> - **error** [string] The error, if any, returned by the run.

post_civis_data_match(*name*, *input_field_mapping*, *input_table*, *match_target_id*, *output_table*, *,
schedule='DEFAULT', *parent_id='DEFAULT'*, *notifications='DEFAULT'*,
max_matches='DEFAULT', *threshold='DEFAULT'*, *archived='DEFAULT'*)

> Create a Civis Data Match Enhancement
>
> > **Parameters**
> >
> > > **name** [string] The name of the enhancement job.
> > >
> > > **input_field_mapping** [dict] The field (i.e., column) mapping for the input table.
> > > See https://api.civisanalytics.com/enhancements/field-mapping for a list of valid
> > > field types and descriptions. Each field type should be mapped to a string specify-
> > > ing a column name in the input table. For field types that support multiple values
> > > (e.g., the "phone" field), a list of column names can be provided (e.g., {"phone":
> > > ["home_phone", "mobile_phone"], … }).
> > >
> > > **input_table** [dict::]
> > >
> > > > - **database_name** [string] The Redshift database name for the table.
> > > > - **schema** [string] The schema name for the table.
> > > > - **table** [string] The table name.
> > >
> > > **match_target_id** [integer] The ID of the Civis Data match target. See /match_targets
> > > for IDs.
> > >
> > > **output_table** [dict::]
> > >
> > > > - **database_name** [string] The Redshift database name for the table.
> > > > - **schema** [string] The schema name for the table.
> > > > - **table** [string] The table name.
> > >
> > > **schedule** [dict, optional::]
> > >
> > > > - **scheduled** [boolean] If the item is scheduled.
> > > > - **scheduled_days** [list] Days of the week, based on numeric value starting
> > > >   at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > > > - **scheduled_hours** [list] Hours of the day it is scheduled on.
> > > > - **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> > > > - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes,
> > > >   number of times to run per hour.
> > > > - **scheduled_days_of_month** [list] Days of the month it is scheduled on,
> > > >   mutually exclusive with scheduledDays.
> > >
> > > **parent_id** [integer, optional] Parent ID that triggers this enhancement.
> > >
> > > **notifications** [dict, optional::]
> > >
> > > > - **urls** [list] URLs to receive a POST request at job completion
> > > > - **success_email_subject** [string] Custom subject line for success e-mail.
> > > > - **success_email_body** [string] Custom body text for success e-mail, writ-
> > > >   ten in Markdown.
> > > > - **success_email_addresses** [list] Addresses to notify by e-mail when the
> > > >   job completes successfully.
> > > > - **success_email_from_name** [string] Name from which success emails are
> > > >   sent; defaults to "Civis."
> > > > - **success_email_reply_to** [string] Address for replies to success emails;
> > > >   defaults to the author of the job.
> > > > - **failure_email_addresses** [list] Addresses to notify by e-mail when the job
> > > >   fails.
> > > > - **stall_warning_minutes** [integer] Stall warning emails will be sent after
> > > >   this amount of minutes.
> > > > - **success_on** [boolean] If success email notifications are on.
> > > > - **failure_on** [boolean] If failure email notifications are on.
> > >
> > > **max_matches** [integer, optional] The maximum number of matches per record in the

input table to return. Must be between 0 and 10. 0 returns all matches.

threshold [number/float, optional] The score threshold (between 0 and 1). Matches below this threshold will not be returned. The default value is 0.5.

archived [boolean, optional] Whether the Civis Data Match Job has been archived.

Returns

    civis.response.Response

- **id** [integer] The ID for the enhancement.
- **name** [string] The name of the enhancement job.
- **type** [string] The type of the enhancement (e.g CASS-NCOA)
- **created_at** [string/time] The time this enhancement was created.
- **updated_at** [string/time] The time the enhancement was last updated.
- **author** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **state** [string] The status of the enhancement's last run
- **schedule** [dict::]

  - **scheduled** [boolean] If the item is scheduled.

  - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  - **scheduled_hours** [list] Hours of the day it is scheduled on.

  - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

  - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

  - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **parent_id** [integer] Parent ID that triggers this enhancement.
- **notifications** [dict::]

  - **urls** [list] URLs to receive a POST request at job completion

  - **success_email_subject** [string] Custom subject line for success e-mail.

  - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

  - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

  - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

  - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **input_field_mapping** [dict] The field (i.e., column) mapping for the input table. See https://api.civisanalytics.com/enhancements/ field-mapping for a list of valid field types and descriptions. Each field type should be mapped to a string specifying a column name in the input table. For field types that support multiple values (e.g., the "phone" field), a list of column names can be provided (e.g., {"phone": ["home_phone", "mobile_phone"], … }).
- **input_table** [dict::]

  - **database_name** [string] The Redshift database name for the table.

  - **schema** [string] The schema name for the table.

  - **table** [string] The table name.
- **match_target_id** [integer] The ID of the Civis Data match target. See /match_targets for IDs.
- **output_table** [dict::]

  - **database_name** [string] The Redshift database name for the table.

  - **schema** [string] The schema name for the table.

  - **table** [string] The table name.
- **max_matches** [integer] The maximum number of matches per record in the input table to return. Must be between 0 and 10. 0 returns all matches.
- **threshold** [number/float] The score threshold (between 0 and 1). Matches below this threshold will not be returned. The default value is 0.5.
- **archived** [boolean] Whether the Civis Data Match Job has been archived.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

post_civis_data_match_cancel(*id*)

> Cancel a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the job.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **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_civis_data_match_clone(*id*, *, *clone_schedule='DEFAULT'*, *clone_triggers='DEFAULT'*, *clone_notifications='DEFAULT'*)

> Clone this Civis Data Match Enhancement
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the enhancement.
> > >
> > > **clone_schedule** [boolean, optional] If true, also copy the schedule to the new enhancement.
> > >
> > > **clone_triggers** [boolean, optional] If true, also copy the triggers to the new enhancement.
> > >
> > > **clone_notifications** [boolean, optional] If true, also copy the notifications to the new enhancement.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **id** [integer] The ID for the enhancement.
> > > > • **name** [string] The name of the enhancement job.
> > > > • **type** [string] The type of the enhancement (e.g CASS-NCOA)
> > > > • **created_at** [string/time] The time this enhancement was created.
> > > > • **updated_at** [string/time] The time the enhancement was last updated.
> > > > • **author** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > • **state** [string] The status of the enhancement's last run
> > > > • **schedule** [dict::]
> > > >
> > > > > – **scheduled** [boolean] If the item is scheduled.
> > > > >
> > > > > – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > > > >
> > > > > – **scheduled_hours** [list] Hours of the day it is scheduled on.
> > > > >
> > > > > – **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> > > > >
> > > > > – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
> > > > >
> > > > > – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
> > > >
> > > > • **parent_id** [integer] Parent ID that triggers this enhancement.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

- **input_field_mapping** [dict] The field (i.e., column) mapping for the input table. See https://api.civisanalytics.com/enhancements/ field-mapping for a list of valid field types and descriptions. Each field type should be mapped to a string specifying a column name in the input table. For field types that support multiple values (e.g., the "phone" field), a list of column names can be provided (e.g., {"phone": ["home_phone", "mobile_phone"], ... }).

- **input_table** [dict::]

    - **database_name** [string] The Redshift database name for the table.

    - **schema** [string] The schema name for the table.

    - **table** [string] The table name.

- **match_target_id** [integer] The ID of the Civis Data match target. See /match_targets for IDs.

- **output_table** [dict::]

    - **database_name** [string] The Redshift database name for the table.

    - **schema** [string] The schema name for the table.

> > > > – **table** [string] The table name.
> > >
> > > - **max_matches** [integer] The maximum number of matches per record in the input table to return. Must be between 0 and 10. 0 returns all matches.
> > > - **threshold** [number/float] The score threshold (between 0 and 1). Matches below this threshold will not be returned. The default value is 0.5.
> > > - **archived** [boolean] Whether the Civis Data Match Job has been archived.
> > > - **last_run** [dict::]
> > >
> > > > > – id : integer
> > > >
> > > > > – state : string
> > > >
> > > > > – **created_at** [string/time] The time that the run was queued.
> > > >
> > > > > – **started_at** [string/time] The time that the run started.
> > > >
> > > > > – **finished_at** [string/time] The time that the run completed.
> > > >
> > > > > – **error** [string] The error message for this run, if present.

post_civis_data_match_runs(*id*)

> Start a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the civis_data_match.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > - **id** [integer] The ID of the run.
> > > - **civis_data_match_id** [integer] The ID of the civis_data_match.
> > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > - **started_at** [string/time] The time the last run started at.
> > > - **finished_at** [string/time] The time the last run completed.
> > > - **error** [string] The error, if any, returned by the run.

post_geocode(*name*, *remote_host_id*, *credential_id*, *source_schema_and_table*, ***, *schedule='DEFAULT'*, *parent_id='DEFAULT'*, *notifications='DEFAULT'*, *multipart_key='DEFAULT'*, *limiting_sql='DEFAULT'*, *target_schema='DEFAULT'*, *target_table='DEFAULT'*, *country='DEFAULT'*, *provider='DEFAULT'*, *output_address='DEFAULT'*)

> Create a Geocode Enhancement
>
> > **Parameters**
> >
> > > **name** [string] The name of the enhancement job.
> > >
> > > **remote_host_id** [integer] The ID of the remote host.
> > >
> > > **credential_id** [integer] The ID of the remote host credential.
> > >
> > > **source_schema_and_table** [string] The source database schema and table.
> > >
> > > **schedule** [dict, optional::]
> > >
> > > > - **scheduled** [boolean] If the item is scheduled.
> > > > - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > > > - **scheduled_hours** [list] Hours of the day it is scheduled on.
> > > > - **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> > > > - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
> > > > - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
> > >
> > > **parent_id** [integer, optional] Parent ID that triggers this enhancement.
> > >
> > > **notifications** [dict, optional::]

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

multipart_key [list, optional] The source table primary key.

limiting_sql [string, optional] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').

target_schema [string, optional] The output table schema.

target_table [string, optional] The output table name.

country [string, optional] The country of the addresses to be geocoded; either 'us' or 'ca'.

provider [string, optional] The geocoding provider; one of postgis, nominatim, and geocoder_ca.

output_address [boolean, optional] Whether to output the parsed address. Only guaranteed for the 'postgis' provider.

Returns

civis.response.Response

- **id** [integer] The ID for the enhancement.
- **name** [string] The name of the enhancement job.
- **type** [string] The type of the enhancement (e.g CASS-NCOA)
- **created_at** [string/time] The time this enhancement was created.
- **updated_at** [string/time] The time the enhancement was last updated.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

- **state** [string] The status of the enhancement's last run
- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

    – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    – **scheduled_hours** [list] Hours of the day it is scheduled on.

    – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **parent_id** [integer] Parent ID that triggers this enhancement.
- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **remote_host_id** [integer] The ID of the remote host.
- **credential_id** [integer] The ID of the remote host credential.
- **source_schema_and_table** [string] The source database schema and table.
- **multipart_key** [list] The source table primary key.
- **limiting_sql** [string] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').
- **target_schema** [string] The output table schema.
- **target_table** [string] The output table name.
- **country** [string] The country of the addresses to be geocoded; either 'us' or 'ca'.
- **provider** [string] The geocoding provider; one of postgis, nominatim, and geocoder_ca.
- **output_address** [boolean] Whether to output the parsed address. Only guaranteed for the 'postgis' provider.
- **archived** [string] The archival status of the requested item(s).

> > • **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

post_geocode_cancel(*id*)

> Cancel a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the job.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **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_geocode_runs(*id*)

> Start a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the geocode.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **id** [integer] The ID of the run.
> > > > • **geocode_id** [integer] The ID of the geocode.
> > > > • **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > > • **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > > • **started_at** [string/time] The time the last run started at.
> > > > • **finished_at** [string/time] The time the last run completed.
> > > > • **error** [string] The error, if any, returned by the run.

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

> Replace all attributes of this CASS/NCOA Enhancement
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the enhancement.
> > >
> > > **name** [string] The name of the enhancement job.
> > >
> > > **source** [dict::]
> > >
> > > > • **database_table** [dict::]
> > > >
> > > > > – **schema** [string] The schema name of the source table.
> > > > >
> > > > > – **table** [string] The name of the source table.
> > > > >
> > > > > – **remote_host_id** [integer] The ID of the database host for the table.
> > > > >
> > > > > – **credential_id** [integer] The id of the credentials to be used when performing the enhancement.
> > > > >
> > > > > – **multipart_key** [list] The source table primary key.
> > >
> > > **schedule** [dict, optional::]
> > >
> > > > • **scheduled** [boolean] If the item is scheduled.
> > > > • **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > > > • **scheduled_hours** [list] Hours of the day it is scheduled on.
> > > > • **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

parent_id [integer, optional] Parent ID that triggers this enhancement.

notifications [dict, optional::]

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

destination [dict, optional::]

- **database_table** [dict::]

    - **schema** [string] The schema name for the output data.

    - **table** [string] The table name for the output data.

column_mapping [dict, optional::]

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

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

perform_ncoa [boolean, optional] Whether to update addresses for records matching the National Change of Address (NCOA) database.

ncoa_credential_id [integer, optional] Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

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

limiting_sql [string, optional] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').

Returns

civis.response.Response

- **id** [integer] The ID for the enhancement.
- **name** [string] The name of the enhancement job.
- **type** [string] The type of the enhancement (e.g CASS-NCOA)

- **created_at** [string/time] The time this enhancement was created.
- **updated_at** [string/time] The time the enhancement was last updated.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **state** [string] The status of the enhancement's last run
- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **parent_id** [integer] Parent ID that triggers this enhancement.
- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]

    - **id** [integer] The ID of this user.

- **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.

- **source** [dict::]

  - **database_table** [dict::]

    * **schema** [string] The schema name of the source table.

    * **table** [string] The name of the source table.

    * **remote_host_id** [integer] The ID of the database host for the table.

    * **credential_id** [integer] The id of the credentials to be used when performing the enhancement.

    * **multipart_key** [list] The source table primary key.

- **destination** [dict::]

  - **database_table** [dict::]

    * **schema** [string] The schema name for the output data.

    * **table** [string] The table name for the output data.

- **column_mapping** [dict::]

  - **address1** [string] The first address line.

  - **address2** [string] The second address line.

  - **city** [string] The city of an address.

  - **state** [string] The state of an address.

  - **zip** [string] The zip code of an address.

  - **name** [string] The full name of the resident at this address. If needed, separate multiple columns with +, e.g. *first_name+last_name*

  - **company** [string] The name of the company located at this address.

- **use_default_column_mapping** [boolean] Defaults to true, where the existing column mapping on the input table will be used. If false, a custom column mapping must be provided.

- **perform_ncoa** [boolean] Whether to update addresses for records matching the National Change of Address (NCOA) database.

- **ncoa_credential_id** [integer] Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

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

- **limiting_sql** [string] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').

- **archived** [string] The archival status of the requested item(s).

> - **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

put_cass_ncoa_archive(*id*, *status*)

> Update the archive status of this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the object.
> > >
> > > **status** [boolean] The desired archived status of the object.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID for the enhancement.
> > > - **name** [string] The name of the enhancement job.
> > > - **type** [string] The type of the enhancement (e.g CASS-NCOA)
> > > - **created_at** [string/time] The time this enhancement was created.
> > > - **updated_at** [string/time] The time the enhancement was last updated.
> > > - **author** [dict::]
> > >
> > > > - **id** [integer] The ID of this user.
> > > >
> > > > - **name** [string] This user's name.
> > > >
> > > > - **username** [string] This user's username.
> > > >
> > > > - **initials** [string] This user's initials.
> > > >
> > > > - **online** [boolean] Whether this user is online.
> > >
> > > - **state** [string] The status of the enhancement's last run
> > > - **schedule** [dict::]
> > >
> > > > - **scheduled** [boolean] If the item is scheduled.
> > > >
> > > > - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > > >
> > > > - **scheduled_hours** [list] Hours of the day it is scheduled on.
> > > >
> > > > - **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> > > >
> > > > - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
> > > >
> > > > - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
> > >
> > > - **parent_id** [integer] Parent ID that triggers this enhancement.
> > > - **notifications** [dict::]
> > >
> > > > - **urls** [list] URLs to receive a POST request at job completion
> > > >
> > > > - **success_email_subject** [string] Custom subject line for success e-mail.
> > > >
> > > > - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.
> > > >
> > > > - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.
> > > >
> > > > - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

- **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **source** [dict::]

    - **database_table** [dict::]

        * **schema** [string] The schema name of the source table.

        * **table** [string] The name of the source table.

        * **remote_host_id** [integer] The ID of the database host for the table.

        * **credential_id** [integer] The id of the credentials to be used when performing the enhancement.

        * **multipart_key** [list] The source table primary key.

- **destination** [dict::]

    - **database_table** [dict::]

        * **schema** [string] The schema name for the output data.

        * **table** [string] The table name for the output data.

- **column_mapping** [dict::]

    - **address1** [string] The first address line.

    - **address2** [string] The second address line.

    - **city** [string] The city of an address.

    - **state** [string] The state of an address.

    - **zip** [string] The zip code of an address.

    - **name** [string] The full name of the resident at this address. If needed, separate multiple columns with +, e.g. *first_name+last_name*

    - **company** [string] The name of the company located at this address.

- **use_default_column_mapping** [boolean] Defaults to true, where the existing column mapping on the input table will be used. If false, a custom column mapping must be provided.
- **perform_ncoa** [boolean] Whether to update addresses for records matching the National Change of Address (NCOA) database.
- **ncoa_credential_id** [integer] Credential to use when performing NCOA updates. Required if 'performNcoa' is true.
- **output_level** [string] The set of fields persisted by a CASS or NCOA enhancement.For CASS enhancements, one of 'cass' or 'all.'For NCOA enhancements, one of 'cass', 'ncoa' , 'coalesced' or 'all'.By default, all fields will be returned.
- **limiting_sql** [string] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').
- **archived** [string] The archival status of the requested item(s).
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

put_cass_ncoa_projects(*id*, *project_id*)

> Add a CASS/NCOA Enhancement to a project
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the CASS/NCOA Enhancement.
> > >
> > > **project_id** [integer] The ID of the project.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

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

> Set the permissions groups has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **group_ids** [list] An array of one or more group IDs.
> > >
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > >
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > >
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> >
> > > > - **readers** [dict::]
> > > >
> > > > > - **users** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > > >
> > > > > - **groups** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > >
> > > > - **writers** [dict::]
> > > >
> > > > > - **users** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > > >
> > > > > - **groups** [list::]
> > > > >
> > > > > > * id : integer

> > > > > > > > > ∗ name : string
> > > > > > - **owners** [dict::]
> > > > > > > > > – **users** [list::]
> > > > > > > > > > > > ∗ id : integer
> > > > > > > > > > > > ∗ name : string
> > > > > > > > > – **groups** [list::]
> > > > > > > > > > > > ∗ id : integer
> > > > > > > > > > > > ∗ name : string
> > > > > > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > > > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_ids** [list] An array of one or more user IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > [*civis.response.Response*](#)
> >
> > > > - **readers** [dict::]
> > > > > > – **users** [list::]
> > > > > > > > ∗ id : integer
> > > > > > > > ∗ name : string
> > > > > > – **groups** [list::]
> > > > > > > > ∗ id : integer
> > > > > > > > ∗ name : string
> > > > - **writers** [dict::]
> > > > > > – **users** [list::]
> > > > > > > > ∗ id : integer
> > > > > > > > ∗ name : string
> > > > > > – **groups** [list::]
> > > > > > > > ∗ id : integer
> > > > > > > > ∗ name : string
> > > > - **owners** [dict::]
> > > > > > – **users** [list::]
> > > > > > > > ∗ id : integer
> > > > > > > > ∗ name : string
> > > > > > – **groups** [list::]

 * id : integer

 * name : string

 - **total_user_shares** [integer] For owners, the number of total users shared.
   For writers and readers, the number of visible users shared.

 - **total_group_shares** [integer] For owners, the number of total groups
   shared. For writers and readers, the number of visible groups shared.

put_cass_ncoa_transfer(*id*, *user_id*, *include_dependencies*, ***, *email_body='DEFAULT'*,
  send_email='DEFAULT')

Transfer ownership of this object to another user

Parameters

id [integer] The ID of the resource that is shared.

user_id [integer] ID of target user

include_dependencies [boolean] Whether or not to give manage permissions on all
  dependencies

email_body [string, optional] Custom body text for e-mail sent on transfer.

send_email [boolean, optional] Send email to the target user of the transfer?

Returns

[*civis.response.Response*](#)

 - **dependencies** [list::] Dependent objects for this object - object_type :
   string

   Dependent object type

   – **fco_type** [string] Human readable dependent object type

   – **id** [integer] Dependent object ID

   – **name** [string] Dependent object name, or nil if the requesting
     user cannot read this object

   – **permission_level** [string] Permission level of target user (not
     user's groups) for dependent object, or null if no target
     user

   – **shared** [boolean] Whether dependent object was successfully
     shared with target user

put_civis_data_match(*id*, *name*, *input_field_mapping*, *input_table*, *match_target_id*, *output_table*, ***,
  schedule='DEFAULT', *parent_id='DEFAULT'*, *notifications='DEFAULT'*,
  max_matches='DEFAULT', *threshold='DEFAULT'*, *archived='DEFAULT'*)

Replace all attributes of this Civis Data Match Enhancement

Parameters

id [integer] The ID for the enhancement.

name [string] The name of the enhancement job.

input_field_mapping [dict] The field (i.e., column) mapping for the input table.
  See https://api.civisanalytics.com/enhancements/field-mapping for a list of valid
  field types and descriptions. Each field type should be mapped to a string specify-
  ing a column name in the input table. For field types that support multiple values
  (e.g., the "phone" field), a list of column names can be provided (e.g., {"phone":
  ["home_phone", "mobile_phone"], ... }).

input_table [dict::]

 - **database_name** [string] The Redshift database name for the table.
 - **schema** [string] The schema name for the table.
 - **table** [string] The table name.

match_target_id [integer] The ID of the Civis Data match target. See /match_targets
  for IDs.

output_table [dict::]
- **database_name** [string] The Redshift database name for the table.
- **schema** [string] The schema name for the table.
- **table** [string] The table name.

schedule [dict, optional::]
- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

parent_id [integer, optional] Parent ID that triggers this enhancement.

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

max_matches [integer, optional] The maximum number of matches per record in the input table to return. Must be between 0 and 10. 0 returns all matches.

threshold [number/float, optional] The score threshold (between 0 and 1). Matches below this threshold will not be returned. The default value is 0.5.

archived [boolean, optional] Whether the Civis Data Match Job has been archived.

Returns

civis.response.Response
- **id** [integer] The ID for the enhancement.
- **name** [string] The name of the enhancement job.
- **type** [string] The type of the enhancement (e.g CASS-NCOA)
- **created_at** [string/time] The time this enhancement was created.
- **updated_at** [string/time] The time the enhancement was last updated.
- **author** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.
- **state** [string] The status of the enhancement's last run

- **schedule** [dict::]

  - **scheduled** [boolean] If the item is scheduled.

  - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  - **scheduled_hours** [list] Hours of the day it is scheduled on.

  - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

  - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

  - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **parent_id** [integer] Parent ID that triggers this enhancement.
- **notifications** [dict::]

  - **urls** [list] URLs to receive a POST request at job completion

  - **success_email_subject** [string] Custom subject line for success e-mail.

  - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

  - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

  - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

  - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

  - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

  - **success_on** [boolean] If success email notifications are on.

  - **failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **input_field_mapping** [dict] The field (i.e., column) mapping for the input table. See https://api.civisanalytics.com/enhancements/field-mapping for a list of valid field types and descriptions. Each field type should be mapped to a string specifying a column name

in the input table. For field types that support multiple values (e.g., the "phone" field), a list of column names can be provided (e.g., {"phone": ["home_phone", "mobile_phone"], ... }).

- **input_table** [dict::]

    – **database_name** [string] The Redshift database name for the table.

    – **schema** [string] The schema name for the table.

    – **table** [string] The table name.

- **match_target_id** [integer] The ID of the Civis Data match target. See /match_targets for IDs.

- **output_table** [dict::]

    – **database_name** [string] The Redshift database name for the table.

    – **schema** [string] The schema name for the table.

    – **table** [string] The table name.

- **max_matches** [integer] The maximum number of matches per record in the input table to return. Must be between 0 and 10. 0 returns all matches.

- **threshold** [number/float] The score threshold (between 0 and 1). Matches below this threshold will not be returned. The default value is 0.5.

- **archived** [boolean] Whether the Civis Data Match Job has been archived.

- **last_run** [dict::]

    – id : integer

    – state : string

    – **created_at** [string/time] The time that the run was queued.

    – **started_at** [string/time] The time that the run started.

    – **finished_at** [string/time] The time that the run completed.

    – **error** [string] The error message for this run, if present.

put_civis_data_match_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**
> >
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID for the enhancement.
> > > - **name** [string] The name of the enhancement job.
> > > - **type** [string] The type of the enhancement (e.g CASS-NCOA)
> > > - **created_at** [string/time] The time this enhancement was created.
> > > - **updated_at** [string/time] The time the enhancement was last updated.
> > > - **author** [dict::]
> > >
> > >   – **id** [integer] The ID of this user.
> > >
> > >   – **name** [string] This user's name.
> > >
> > >   – **username** [string] This user's username.
> > >
> > >   – **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **state** [string] The status of the enhancement's last run
- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **parent_id** [integer] Parent ID that triggers this enhancement.
- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **input_field_mapping** [dict] The field (i.e., column) mapping for the input table. See https://api.civisanalytics.com/enhancements/

field-mapping for a list of valid field types and descriptions. Each
field type should be mapped to a string specifying a column name
in the input table. For field types that support multiple values (e.g.,
the "phone" field), a list of column names can be provided (e.g.,
{"phone": ["home_phone", "mobile_phone"], ... }).

- **input_table** [dict::]

    – **database_name** [string] The Redshift database name for the
        table.

    – **schema** [string] The schema name for the table.

    – **table** [string] The table name.
- **match_target_id** [integer] The ID of the Civis Data match target. See
    /match_targets for IDs.
- **output_table** [dict::]

    – **database_name** [string] The Redshift database name for the
        table.

    – **schema** [string] The schema name for the table.

    – **table** [string] The table name.
- **max_matches** [integer] The maximum number of matches per record in
    the input table to return. Must be between 0 and 10. 0 returns all
    matches.
- **threshold** [number/float] The score threshold (between 0 and 1). Matches
    below this threshold will not be returned. The default value is 0.5.
- **archived** [boolean] Whether the Civis Data Match Job has been archived.
- **last_run** [dict::]

    – id : integer

    – state : string

    – **created_at** [string/time] The time that the run was queued.

    – **started_at** [string/time] The time that the run started.

    – **finished_at** [string/time] The time that the run completed.

    – **error** [string] The error message for this run, if present.

put_civis_data_match_projects(*id*, *project_id*)

    Add a Civis Data Match Enhancement to a project

        Parameters

            id [integer] The ID of the Civis Data Match Enhancement.

            project_id [integer] The ID of the project.

        Returns

            None Response code 204: success

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

    Set the permissions groups has on this object

        Parameters

            id [integer] The ID of the resource that is shared.

            group_ids [list] An array of one or more group IDs.

            permission_level [string] Options are: "read", "write", or "manage".

            share_email_body [string, optional] Custom body text for e-mail sent on a share.

            send_shared_email [boolean, optional] Send email to the recipients of a share.

> **Returns**
> *civis.response.Response*
> - **readers** [dict::]
>   - **users** [list::]
>   * id : integer
>   * name : string
>   - **groups** [list::]
>   * id : integer
>   * name : string
> - **writers** [dict::]
>   - **users** [list::]
>   * id : integer
>   * name : string
>   - **groups** [list::]
>   * id : integer
>   * name : string
> - **owners** [dict::]
>   - **users** [list::]
>   * id : integer
>   * name : string
>   - **groups** [list::]
>   * id : integer
>   * name : string
> - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object
> > **Parameters**
> > **id** [integer] The ID of the resource that is shared.
> > **user_ids** [list] An array of one or more user IDs.
> > **permission_level** [string] Options are: "read", "write", or "manage".
> > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> > **Returns**
> > *civis.response.Response*
> > - **readers** [dict::]
> >   - **users** [list::]
> >   * id : integer
> >   * name : string

> > > > > – **groups** [list::]

> > > > > > > > * id : integer

> > > > > > > > * name : string

> > > > • **writers** [dict::]

> > > > > – **users** [list::]

> > > > > > > > * id : integer

> > > > > > > > * name : string

> > > > > – **groups** [list::]

> > > > > > > > * id : integer

> > > > > > > > * name : string

> > > > • **owners** [dict::]

> > > > > – **users** [list::]

> > > > > > > > * id : integer

> > > > > > > > * name : string

> > > > > – **groups** [list::]

> > > > > > > > * id : integer

> > > > > > > > * name : string

> > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_civis_data_match_transfer(*id*, *user_id*, *include_dependencies*, ***, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

> Transfer ownership of this object to another user

> > **Parameters**

> > > **id** [integer] The ID of the resource that is shared.
> > > **user_id** [integer] ID of target user
> > > **include_dependencies** [boolean] Whether or not to give manage permissions on all dependencies
> > > **email_body** [string, optional] Custom body text for e-mail sent on transfer.
> > > **send_email** [boolean, optional] Send email to the target user of the transfer?

> > **Returns**

> > > *civis.response.Response*

> > > > • **dependencies** [list::] Dependent objects for this object - object_type : string

> > > > > Dependent object type

> > > > > – **fco_type** [string] Human readable dependent object type

> > > > > – **id** [integer] Dependent object ID

> > > > > – **name** [string] Dependent object name, or nil if the requesting user cannot read this object

> > > > > – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user

– **shared** [boolean] Whether dependent object was successfully
shared with target user

put_geocode(*id*, *name*, *remote_host_id*, *credential_id*, *source_schema_and_table*, *, *schedule='DEFAULT'*,
parent_id='DEFAULT', *notifications='DEFAULT'*, *multipart_key='DEFAULT'*,
limiting_sql='DEFAULT', *target_schema='DEFAULT'*, *target_table='DEFAULT'*,
country='DEFAULT', *provider='DEFAULT'*, *output_address='DEFAULT'*)

Replace all attributes of this Geocode Enhancement

Parameters

id [integer] The ID for the enhancement.

name [string] The name of the enhancement job.

remote_host_id [integer] The ID of the remote host.

credential_id [integer] The ID of the remote host credential.

source_schema_and_table [string] The source database schema and table.

schedule [dict, optional::]
- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting
at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes,
number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on,
mutually exclusive with scheduledDays.

parent_id [integer, optional] Parent ID that triggers this enhancement.

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

multipart_key [list, optional] The source table primary key.

limiting_sql [string, optional] The limiting SQL for the source table. "WHERE"
should be omitted (e.g. state='IL').

target_schema [string, optional] The output table schema.

target_table [string, optional] The output table name.

country [string, optional] The country of the addresses to be geocoded; either 'us' or
'ca'.

provider [string, optional] The geocoding provider; one of postgis, nominatim, and
geocoder_ca.

output_address [boolean, optional] Whether to output the parsed address. Only guar-
anteed for the 'postgis' provider.

Returns

`civis.response.Response`
- **id** [integer] The ID for the enhancement.
- **name** [string] The name of the enhancement job.
- **type** [string] The type of the enhancement (e.g CASS-NCOA)
- **created_at** [string/time] The time this enhancement was created.
- **updated_at** [string/time] The time the enhancement was last updated.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- **state** [string] The status of the enhancement's last run
- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

    – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    – **scheduled_hours** [list] Hours of the day it is scheduled on.

    – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **parent_id** [integer] Parent ID that triggers this enhancement.
- **notifications** [dict::]

    – **urls** [list] URLs to receive a POST request at job completion

    – **success_email_subject** [string] Custom subject line for success e-mail.

    – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    – **success_on** [boolean] If success email notifications are on.

- **– failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]

    - **– id** [integer] The ID of this user.

    - **– name** [string] This user's name.

    - **– username** [string] This user's username.

    - **– initials** [string] This user's initials.

    - **– online** [boolean] Whether this user is online.
- **remote_host_id** [integer] The ID of the remote host.
- **credential_id** [integer] The ID of the remote host credential.
- **source_schema_and_table** [string] The source database schema and table.
- **multipart_key** [list] The source table primary key.
- **limiting_sql** [string] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').
- **target_schema** [string] The output table schema.
- **target_table** [string] The output table name.
- **country** [string] The country of the addresses to be geocoded; either 'us' or 'ca'.
- **provider** [string] The geocoding provider; one of postgis, nominatim, and geocoder_ca.
- **output_address** [boolean] Whether to output the parsed address. Only guaranteed for the 'postgis' provider.
- **archived** [string] The archival status of the requested item(s).
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

put_geocode_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**
> >
> > > [*civis.response.Response*](#)
> > >
> > > - **id** [integer] The ID for the enhancement.
> > > - **name** [string] The name of the enhancement job.
> > > - **type** [string] The type of the enhancement (e.g CASS-NCOA)
> > > - **created_at** [string/time] The time this enhancement was created.
> > > - **updated_at** [string/time] The time the enhancement was last updated.
> > > - **author** [dict::]
> > >
> > >   - **– id** [integer] The ID of this user.
> > >
> > >   - **– name** [string] This user's name.
> > >
> > >   - **– username** [string] This user's username.
> > >
> > >   - **– initials** [string] This user's initials.
> > >
> > >   - **– online** [boolean] Whether this user is online.
> > > - **state** [string] The status of the enhancement's last run
> > > - **schedule** [dict::]
> > >
> > >   - **– scheduled** [boolean] If the item is scheduled.

> – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
>
> – **scheduled_hours** [list] Hours of the day it is scheduled on.
>
> – **scheduled_minutes** [list] Minutes of the day it is scheduled on.
>
> – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
>
> – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **parent_id** [integer] Parent ID that triggers this enhancement.
- **notifications** [dict::]

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

- **running_as** [dict::]

> – **id** [integer] The ID of this user.
>
> – **name** [string] This user's name.
>
> – **username** [string] This user's username.
>
> – **initials** [string] This user's initials.
>
> – **online** [boolean] Whether this user is online.

- **remote_host_id** [integer] The ID of the remote host.
- **credential_id** [integer] The ID of the remote host credential.
- **source_schema_and_table** [string] The source database schema and table.
- **multipart_key** [list] The source table primary key.
- **limiting_sql** [string] The limiting SQL for the source table. "WHERE" should be omitted (e.g. state='IL').
- **target_schema** [string] The output table schema.
- **target_table** [string] The output table name.

> - **country** [string] The country of the addresses to be geocoded; either 'us' or 'ca'.
> - **provider** [string] The geocoding provider; one of postgis, nominatim, and geocoder_ca.
> - **output_address** [boolean] Whether to output the parsed address. Only guaranteed for the 'postgis' provider.
> - **archived** [string] The archival status of the requested item(s).
> - **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

put_geocode_projects(*id*, *project_id*)

> Add a Geocode Enhancement to a project
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the Geocode Enhancement.
> > > **project_id** [integer] The ID of the project.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

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

> Set the permissions groups has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > > **group_ids** [list] An array of one or more group IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > - **readers** [dict::]
> > > >
> > > >   - **users** [list::]
> > > >
> > > >   * id : integer
> > > >
> > > >   * name : string
> > > >
> > > >   - **groups** [list::]
> > > >
> > > >   * id : integer
> > > >
> > > >   * name : string
> > > >
> > > > - **writers** [dict::]
> > > >
> > > >   - **users** [list::]
> > > >
> > > >   * id : integer
> > > >
> > > >   * name : string
> > > >
> > > >   - **groups** [list::]
> > > >
> > > >   * id : integer
> > > >
> > > >   * name : string
> > > >
> > > > - **owners** [dict::]
> > > >
> > > >   - **users** [list::]
> > > >
> > > >   * id : integer
> > > >
> > > >   * name : string

> > > – **groups** [list::]

> > > > * id : integer

> > > > * name : string

> > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_ids** [list] An array of one or more user IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **readers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > >
> > > > • **writers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > >
> > > > • **owners** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > >
> > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_geocode_transfer(*id*, *user_id*, *include_dependencies*, *, *email_body='DEFAULT'*,
send_email='DEFAULT')

> Transfer ownership of this object to another user
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] ID of target user
> > >
> > > **include_dependencies** [boolean] Whether or not to give manage permissions on all
> > > dependencies
> > >
> > > **email_body** [string, optional] Custom body text for e-mail sent on transfer.
> > >
> > > **send_email** [boolean, optional] Send email to the target user of the transfer?
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **dependencies** [list::] Dependent objects for this object - object_type :
> > > > string
> > > >
> > > > > Dependent object type
> > > >
> > > > – **fco_type** [string] Human readable dependent object type
> > > >
> > > > – **id** [integer] Dependent object ID
> > > >
> > > > – **name** [string] Dependent object name, or nil if the requesting
> > > > user cannot read this object
> > > >
> > > > – **permission_level** [string] Permission level of target user (not
> > > > user's groups) for dependent object, or null if no target
> > > > user
> > > >
> > > > – **shared** [boolean] Whether dependent object was successfully
> > > > shared with target user

Exports

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

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.exports.list(...)
```

Methods

delete_files_csv_runs(id, run_id)	Cancel a run
get_files_csv(id)	Get a CSV Export
get_files_csv_runs(id, run_id)	Check status of a run
list(*[, type, status, author, hidden, ...])	List
list_files_csv_runs(id, *[, limit, ...])	List runs for the given csv_export
list_files_csv_runs_logs(id, run_id, *[, ...])	Get the logs for a run
list_files_csv_runs_outputs(id, run_id, *[, ...])	List the outputs for a run

continues on next page

Table 23 – continued from previous page

patch_files_csv(id, *[, name, source, ...])	Update some attributes of this CSV Export
post_files_csv(source, destination, *[, ...])	Create a CSV Export
post_files_csv_runs(id)	Start a run
put_files_csv(id, source, destination, *[, ...])	Replace all attributes of this CSV Export
put_files_csv_archive(id, status)	Update the archive status of this object

—

delete_files_csv_runs(*id*, *run_id*)

> Cancel a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the csv_export.
> > >
> > > **run_id** [integer] The ID of the run.
> >
> > **Returns**
> >
> > > **None** Response code 202: success

get_files_csv(*id*)

> Get a CSV Export
>
> > **Parameters**
> >
> > > **id** [integer]
> >
> > **Returns**
> >
> > > *civis.response.Response*
> >
> > > > • **id** [integer] The ID of this Csv Export job.
> > > >
> > > > • **name** [string] The name of this Csv Export job.
> > > >
> > > > • **source** [dict::]
> > > >
> > > > > – **sql** [string] The SQL query for this Csv Export job
> > > > >
> > > > > – **remote_host_id** [integer] The ID of the destination database host.
> > > > >
> > > > > – **credential_id** [integer] The ID of the credentials for the destination database.
> > > >
> > > > • **destination** [dict::]
> > > >
> > > > > – **filename_prefix** [string] The prefix of the name of the file returned to the user.
> > > > >
> > > > > – **storage_path** [dict::]
> > > > >
> > > > > > ∗ **file_path** [string] The path within the bucket where the exported file will be saved. E.g. the file_path for "s3://mybucket/files/all/" would be "/files/all/"
> > > > > >
> > > > > > ∗ **storage_host_id** [integer] The ID of the destination storage host.
> > > > > >
> > > > > > ∗ **credential_id** [integer] The ID of the credentials for the destination storage host.
> > > > > >
> > > > > > ∗ **existing_files** [string] Notifies the job of what to do in the case that the exported file already exists at the provided path.One of: fail, append, overwrite. Default: fail. If "append" is specified,the new file will always be added to the provided path. If "overwrite" is specifiedall existing files

> at the provided path will be deleted and the new
> file will be added.By default, or if "fail" is spec-
> ified, the export will fail if a file exists at the
> provided path.

- **include_header** [boolean] A boolean value indicating whether or not the header should be included. Defaults to true.
- **compression** [string] The compression of the output file. Valid arguments are "gzip" and "none". Defaults to "gzip".
- **column_delimiter** [string] The column delimiter for the output file. Valid arguments are "comma", "tab", and "pipe". Defaults to "comma".
- **hidden** [boolean] A boolean value indicating whether or not this request should be hidden. Defaults to false.
- **force_multifile** [boolean] Whether or not the csv should be split into multiple files. Default: false
- **max_file_size** [integer] The max file size, in MB, created files will be. Only available when force_multifile is true.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

get_files_csv_runs(*id*, *run_id*)

> Check status of a run
>> **Parameters**
>>> **id** [integer] The ID of the csv_export.
>>> **run_id** [integer] The ID of the run.
>> **Returns**
>>> *civis.response.Response*
>>>> - id : integer
>>>> - state : string
>>>> - **created_at** [string/time] The time that the run was queued.
>>>> - **started_at** [string/time] The time that the run started.
>>>> - **finished_at** [string/time] The time that the run completed.
>>>> - **error** [string] The error message for this run, if present.
>>>> - **output_cached_on** [string/time] The time that the output was originally exported, if a cache entry was used by the run.

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

> List
>> **Parameters**
>>> **type** [string, optional] If specified, return exports of these types. It accepts a comma-separated list, possible values are 'database' and 'gdoc'.
>>> **status** [string, optional] If specified, returns export with one of these statuses. It accepts a comma-separated list, possible values are 'running', 'failed', 'succeeded', 'idle', 'scheduled'.
>>> **author** [string, optional] If specified, return items from any of these authors. It accepts a comma- separated list of user IDs.
>>> **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
>>> **archived** [string, optional] The archival status of the requested item(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 up-

> dated_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**
>
> > *civis.response.PaginatedResponse*
> >
> > > - **id** [integer] The ID for this export.
> > > - **name** [string] The name of this export.
> > > - **type** [string] The type of export.
> > > - **created_at** [string/time] The creation time for this export.
> > > - **updated_at** [string/time] The last modification time for this export.
> > > - state : string
> > > - **last_run** [dict::]
> > >
> > > > - id : integer
> > > >
> > > > - state : string
> > > >
> > > > - **created_at** [string/time] The time that the run was queued.
> > > >
> > > > - **started_at** [string/time] The time that the run started.
> > > >
> > > > - **finished_at** [string/time] The time that the run completed.
> > > >
> > > > - **error** [string] The error message for this run, if present.
> > >
> > > - **author** [dict::]
> > >
> > > > - **id** [integer] The ID of this user.
> > > >
> > > > - **name** [string] This user's name.
> > > >
> > > > - **username** [string] This user's username.
> > > >
> > > > - **initials** [string] This user's initials.
> > > >
> > > > - **online** [boolean] Whether this user is online.

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

> List runs for the given csv_export
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the csv_export.
> > >
> > > **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**
> >
> > > *civis.response.PaginatedResponse*
> > >
> > > > - id : integer
> > > > - state : string
> > > > - **created_at** [string/time] The time that the run was queued.

> > > > - **started_at** [string/time] The time that the run started.
> > > > - **finished_at** [string/time] The time that the run completed.
> > > > - **error** [string] The error message for this run, if present.

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

> Get the logs for a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the csv_export.
> > >
> > > **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**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > - **id** [integer] The ID of the log.
> > > > - **created_at** [string/date-time] The time the log was created.
> > > > - **message** [string] The log message.
> > > > - **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

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

> List the outputs for a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the csv_export.
> > >
> > > **run_id** [integer] The ID of the run.
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to its maximum of 50.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to created_at. Must be one of: created_at, id.
> > >
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > >
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> >
> > **Returns**
> >
> > > [*civis.response.PaginatedResponse*](#)
> > >
> > > > - **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
> > > > - **object_id** [integer] The ID of the output.
> > > > - **name** [string] The name of the output.
> > > > - **link** [string] The hypermedia link to the output.
> > > > - value : string

patch_files_csv(*id*, *, *name='DEFAULT'*, *source='DEFAULT'*, *destination='DEFAULT'*, *include_header='DEFAULT'*, *compression='DEFAULT'*, *column_delimiter='DEFAULT'*, *hidden='DEFAULT'*, *force_multifile='DEFAULT'*, *max_file_size='DEFAULT'*)

> Update some attributes of this CSV Export
>
> > **Parameters**
> >
> > > **id** [integer] The ID of this Csv Export job.
> > >
> > > **name** [string, optional] The name of this Csv Export job.
> > >
> > > **source** [dict, optional::]

- **sql** [string] The SQL query for this Csv Export job
- **remote_host_id** [integer] The ID of the destination database host.
- **credential_id** [integer] The ID of the credentials for the destination database.

destination [dict, optional::]

- **filename_prefix** [string] The prefix of the name of the file returned to the user.
- **storage_path** [dict::]

    – **file_path** [string] The path within the bucket where the exported file will be saved. E.g. the file_path for "s3://mybucket/files/all/" would be "/files/all/"

    – **storage_host_id** [integer] The ID of the destination storage host.

    – **credential_id** [integer] The ID of the credentials for the destination storage host.

    – **existing_files** [string] Notifies the job of what to do in the case that the exported file already exists at the provided path.One of: fail, append, overwrite. Default: fail. If "append" is specified,the new file will always be added to the provided path. If "overwrite" is specifiedall existing files at the provided path will be deleted and the new file will be added.By default, or if "fail" is specified, the export will fail if a file exists at the provided path.

include_header [boolean, optional] A boolean value indicating whether or not the header should be included. Defaults to true.

compression [string, optional] The compression of the output file. Valid arguments are "gzip" and "none". Defaults to "gzip".

column_delimiter [string, optional] The column delimiter for the output file. Valid arguments are "comma", "tab", and "pipe". Defaults to "comma".

hidden [boolean, optional] A boolean value indicating whether or not this request should be hidden. Defaults to false.

force_multifile [boolean, optional] Whether or not the csv should be split into multiple files. Default: false

max_file_size [integer, optional] The max file size, in MB, created files will be. Only available when force_multifile is true.

Returns

civis.response.Response

- **id** [integer] The ID of this Csv Export job.
- **name** [string] The name of this Csv Export job.
- **source** [dict::]

    – **sql** [string] The SQL query for this Csv Export job

    – **remote_host_id** [integer] The ID of the destination database host.

    – **credential_id** [integer] The ID of the credentials for the destination database.

- **destination** [dict::]

    – **filename_prefix** [string] The prefix of the name of the file returned to the user.

    – **storage_path** [dict::]

> > > > ∗ **file_path** [string] The path within the bucket where
> > > > the exported file will be saved. E.g. the
> > > > file_path for "s3://mybucket/files/all/" would be
> > > > "/files/all/"
> > > >
> > > > ∗ **storage_host_id** [integer] The ID of the destina-
> > > > tion storage host.
> > > >
> > > > ∗ **credential_id** [integer] The ID of the credentials
> > > > for the destination storage host.
> > > >
> > > > ∗ **existing_files** [string] Notifies the job of what to do
> > > > in the case that the exported file already exists
> > > > at the provided path.One of: fail, append, over-
> > > > write. Default: fail. If "append" is specified,the
> > > > new file will always be added to the provided
> > > > path. If "overwrite" is specifiedall existing files
> > > > at the provided path will be deleted and the new
> > > > file will be added.By default, or if "fail" is spec-
> > > > ified, the export will fail if a file exists at the
> > > > provided path.
> > >
> > > • **include_header** [boolean] A boolean value indicating whether or not the
> > > header should be included. Defaults to true.
> > > • **compression** [string] The compression of the output file. Valid arguments
> > > are "gzip" and "none". Defaults to "gzip".
> > > • **column_delimiter** [string] The column delimiter for the output file. Valid
> > > arguments are "comma", "tab", and "pipe". Defaults to "comma".
> > > • **hidden** [boolean] A boolean value indicating whether or not this request
> > > should be hidden. Defaults to false.
> > > • **force_multifile** [boolean] Whether or not the csv should be split into mul-
> > > tiple files. Default: false
> > > • **max_file_size** [integer] The max file size, in MB, created files will be.
> > > Only available when force_multifile is true.
> > > • **my_permission_level** [string] Your permission level on the object. One
> > > of "read", "write", or "manage".

post_files_csv(*source*, *destination*, *, *name='DEFAULT'*, *include_header='DEFAULT'*,
compression='DEFAULT', *column_delimiter='DEFAULT'*, *hidden='DEFAULT'*,
force_multifile='DEFAULT', *max_file_size='DEFAULT'*)

> Create a CSV Export
>
> > **Parameters**
> >
> > > **source** [dict::]
> > > • **sql** [string] The SQL query for this Csv Export job
> > > • **remote_host_id** [integer] The ID of the destination database host.
> > > • **credential_id** [integer] The ID of the credentials for the destination
> > > database.
> > >
> > > **destination** [dict::]
> > > • **filename_prefix** [string] The prefix of the name of the file returned to the
> > > user.
> > > • **storage_path** [dict::]
> > >
> > > > – **file_path** [string] The path within the bucket where the ex-
> > > > ported file will be saved. E.g. the file_path for
> > > > "s3://mybucket/files/all/" would be "/files/all/"
> > > >
> > > > – **storage_host_id** [integer] The ID of the destination storage
> > > > host.

> – **credential_id** [integer] The ID of the credentials for the destination storage host.
>
> – **existing_files** [string] Notifies the job of what to do in the case that the exported file already exists at the provided path.One of: fail, append, overwrite. Default: fail. If "append" is specified,the new file will always be added to the provided path. If "overwrite" is specifiedall existing files at the provided path will be deleted and the new file will be added.By default, or if "fail" is specified, the export will fail if a file exists at the provided path.

name [string, optional] The name of this Csv Export job.

include_header [boolean, optional] A boolean value indicating whether or not the header should be included. Defaults to true.

compression [string, optional] The compression of the output file. Valid arguments are "gzip" and "none". Defaults to "gzip".

column_delimiter [string, optional] The column delimiter for the output file. Valid arguments are "comma", "tab", and "pipe". Defaults to "comma".

hidden [boolean, optional] A boolean value indicating whether or not this request should be hidden. Defaults to false.

force_multifile [boolean, optional] Whether or not the csv should be split into multiple files. Default: false

max_file_size [integer, optional] The max file size, in MB, created files will be. Only available when force_multifile is true.

Returns

civis.response.Response

- **id** [integer] The ID of this Csv Export job.
- **name** [string] The name of this Csv Export job.
- **source** [dict::]

> – **sql** [string] The SQL query for this Csv Export job
>
> – **remote_host_id** [integer] The ID of the destination database host.
>
> – **credential_id** [integer] The ID of the credentials for the destination database.

- **destination** [dict::]

> – **filename_prefix** [string] The prefix of the name of the file returned to the user.
>
> – **storage_path** [dict::]
>
> > ∗ **file_path** [string] The path within the bucket where the exported file will be saved. E.g. the file_path for "s3://mybucket/files/all/" would be "/files/all/"
> >
> > ∗ **storage_host_id** [integer] The ID of the destination storage host.
> >
> > ∗ **credential_id** [integer] The ID of the credentials for the destination storage host.
> >
> > ∗ **existing_files** [string] Notifies the job of what to do in the case that the exported file already exists at the provided path.One of: fail, append, overwrite. Default: fail. If "append" is specified,the

> > > new file will always be added to the provided path. If "overwrite" is specifiedall existing files at the provided path will be deleted and the new file will be added.By default, or if "fail" is specified, the export will fail if a file exists at the provided path.

- **include_header** [boolean] A boolean value indicating whether or not the header should be included. Defaults to true.
- **compression** [string] The compression of the output file. Valid arguments are "gzip" and "none". Defaults to "gzip".
- **column_delimiter** [string] The column delimiter for the output file. Valid arguments are "comma", "tab", and "pipe". Defaults to "comma".
- **hidden** [boolean] A boolean value indicating whether or not this request should be hidden. Defaults to false.
- **force_multifile** [boolean] Whether or not the csv should be split into multiple files. Default: false
- **max_file_size** [integer] The max file size, in MB, created files will be. Only available when force_multifile is true.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

post_files_csv_runs(*id*)

> Start a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the csv_export.
> >
> > **Returns**
> >
> > [*civis.response.Response*](#)
> >
> > > - id : integer
> > > - state : string
> > > - **created_at** [string/time] The time that the run was queued.
> > > - **started_at** [string/time] The time that the run started.
> > > - **finished_at** [string/time] The time that the run completed.
> > > - **error** [string] The error message for this run, if present.
> > > - **output_cached_on** [string/time] The time that the output was originally exported, if a cache entry was used by the run.

put_files_csv(*id*, *source*, *destination*, ***, *name='DEFAULT'*, *include_header='DEFAULT'*, *compression='DEFAULT'*, *column_delimiter='DEFAULT'*, *hidden='DEFAULT'*, *force_multifile='DEFAULT'*, *max_file_size='DEFAULT'*)

> Replace all attributes of this CSV Export
>
> > **Parameters**
> >
> > > **id** [integer] The ID of this Csv Export job.
> > >
> > > **source** [dict::]
> > >
> > > > - **sql** [string] The SQL query for this Csv Export job
> > > > - **remote_host_id** [integer] The ID of the destination database host.
> > > > - **credential_id** [integer] The ID of the credentials for the destination database.
> > >
> > > **destination** [dict::]
> > >
> > > > - **filename_prefix** [string] The prefix of the name of the file returned to the user.
> > > > - **storage_path** [dict::]
> > > >
> > > > > – **file_path** [string] The path within the bucket where the exported file will be saved. E.g. the file_path for "s3://mybucket/files/all/" would be "/files/all/"

> – **storage_host_id** [integer] The ID of the destination storage host.
>
> – **credential_id** [integer] The ID of the credentials for the destination storage host.
>
> – **existing_files** [string] Notifies the job of what to do in the case that the exported file already exists at the provided path.One of: fail, append, overwrite. Default: fail. If "append" is specified,the new file will always be added to the provided path. If "overwrite" is specifiedall existing files at the provided path will be deleted and the new file will be added.By default, or if "fail" is specified, the export will fail if a file exists at the provided path.

name [string, optional] The name of this Csv Export job.

include_header [boolean, optional] A boolean value indicating whether or not the header should be included. Defaults to true.

compression [string, optional] The compression of the output file. Valid arguments are "gzip" and "none". Defaults to "gzip".

column_delimiter [string, optional] The column delimiter for the output file. Valid arguments are "comma", "tab", and "pipe". Defaults to "comma".

hidden [boolean, optional] A boolean value indicating whether or not this request should be hidden. Defaults to false.

force_multifile [boolean, optional] Whether or not the csv should be split into multiple files. Default: false

max_file_size [integer, optional] The max file size, in MB, created files will be. Only available when force_multifile is true.

Returns

[*civis.response.Response*](#)

- **id** [integer] The ID of this Csv Export job.
- **name** [string] The name of this Csv Export job.
- **source** [dict::]

    – **sql** [string] The SQL query for this Csv Export job

    – **remote_host_id** [integer] The ID of the destination database host.

    – **credential_id** [integer] The ID of the credentials for the destination database.

- **destination** [dict::]

    – **filename_prefix** [string] The prefix of the name of the file returned to the user.

    – **storage_path** [dict::]

        * **file_path** [string] The path within the bucket where the exported file will be saved. E.g. the file_path for "s3://mybucket/files/all/" would be "/files/all/"

        * **storage_host_id** [integer] The ID of the destination storage host.

        * **credential_id** [integer] The ID of the credentials for the destination storage host.

        * **existing_files** [string] Notifies the job of what to do

---

> in the case that the exported file already exists at the provided path.One of: fail, append, overwrite. Default: fail. If "append" is specified,the new file will always be added to the provided path. If "overwrite" is specifiedall existing files at the provided path will be deleted and the new file will be added.By default, or if "fail" is specified, the export will fail if a file exists at the provided path.

- **include_header** [boolean] A boolean value indicating whether or not the header should be included. Defaults to true.
- **compression** [string] The compression of the output file. Valid arguments are "gzip" and "none". Defaults to "gzip".
- **column_delimiter** [string] The column delimiter for the output file. Valid arguments are "comma", "tab", and "pipe". Defaults to "comma".
- **hidden** [boolean] A boolean value indicating whether or not this request should be hidden. Defaults to false.
- **force_multifile** [boolean] Whether or not the csv should be split into multiple files. Default: false
- **max_file_size** [integer] The max file size, in MB, created files will be. Only available when force_multifile is true.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

put_files_csv_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**
>> [*civis.response.Response*](#)
>>
>> - **id** [integer] The ID of this Csv Export job.
>> - **name** [string] The name of this Csv Export job.
>> - **source** [dict::]
>>
>>> – **sql** [string] The SQL query for this Csv Export job
>>>
>>> – **remote_host_id** [integer] The ID of the destination database host.
>>>
>>> – **credential_id** [integer] The ID of the credentials for the destination database.
>>
>> - **destination** [dict::]
>>
>>> – **filename_prefix** [string] The prefix of the name of the file returned to the user.
>>>
>>> – **storage_path** [dict::]
>>>
>>>> * **file_path** [string] The path within the bucket where the exported file will be saved. E.g. the file_path for "s3://mybucket/files/all/" would be "/files/all/"
>>>>
>>>> * **storage_host_id** [integer] The ID of the destination storage host.
>>>>
>>>> * **credential_id** [integer] The ID of the credentials

for the destination storage host.

* **existing_files** [string] Notifies the job of what to do in the case that the exported file already exists at the provided path.One of: fail, append, overwrite. Default: fail. If "append" is specified,the new file will always be added to the provided path. If "overwrite" is specifiedall existing files at the provided path will be deleted and the new file will be added.By default, or if "fail" is specified, the export will fail if a file exists at the provided path.

- **include_header** [boolean] A boolean value indicating whether or not the header should be included. Defaults to true.
- **compression** [string] The compression of the output file. Valid arguments are "gzip" and "none". Defaults to "gzip".
- **column_delimiter** [string] The column delimiter for the output file. Valid arguments are "comma", "tab", and "pipe". Defaults to "comma".
- **hidden** [boolean] A boolean value indicating whether or not this request should be hidden. Defaults to false.
- **force_multifile** [boolean] Whether or not the csv should be split into multiple files. Default: false
- **max_file_size** [integer] The max file size, in MB, created files will be. Only available when force_multifile is true.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

Files

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

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.files.list_projects(...)
```

Methods

delete_projects(id, project_id)	Remove a File 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, *[, link_expires_at, inline])	Get details about a file
get_preprocess_csv(id)	Get a Preprocess CSV
list_dependencies(id, *[, user_id])	List dependent objects for this object
list_projects(id, *[, hidden])	List the projects a File belongs to
list_shares(id)	List users and groups permissioned on this object
patch(id, *[, name, expires_at])	Update details about a file

continues on next page

Table 25 – continued from previous page

patch_preprocess_csv(id, *[, file_id, ...])	Update some attributes of this Preprocess CSV
post(name, *[, expires_at])	Initiate an upload of a file into the platform
post_multipart(name, num_parts, *[, expires_at])	Initiate a multipart upload
post_multipart_complete(id)	Complete a multipart upload
post_preprocess_csv(file_id, *[, in_place, ...])	Create a Preprocess CSV
put(id, name, expires_at)	Update details about a file
put_preprocess_csv(id, file_id, *[, ...])	Replace all attributes of this Preprocess CSV
put_preprocess_csv_archive(id, status)	Update the archive status of this object
put_projects(id, project_id)	Add a File to a project
put_shares_groups(id, group_ids, ...[, ...])	Set the permissions groups has on this object
put_shares_users(id, user_ids, ...[, ...])	Set the permissions users have on this object
put_transfer(id, user_id, ...[, email_body, ...])	Transfer ownership of this object to another user

———

delete_projects(*id*, *project_id*)

Remove a File from a project

> **Parameters**
>> **id** [integer] The ID of the File.
>> **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] The ID of the resource that is shared.
>> **group_id** [integer] The 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] The ID of the resource that is shared.
>> **user_id** [integer] The ID of the user.
>
> **Returns**
>> **None** Response code 204: success

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

Get details about a file

> **Parameters**
>> **id** [integer] The ID of the file.
>> **link_expires_at** [string, optional] The date and time the download link will expire. Must be a time between now and 36 hours from now. Defaults to 30 minutes from now.
>> **inline** [boolean, optional] If true, will return a url that can be displayed inline in HTML
>
> **Returns**
>> [civis.response.Response](#)
>>> • **id** [integer] The ID of the file.
>>> • **name** [string] The file name.
>>> • **created_at** [string/date-time] The date and time the file was created.
>>> • **file_size** [integer] The file size.

- **expires_at** [string/date-time] The date and time the file will expire. If not specified, the file will expire in 30 days. To keep a file indefinitely, specify null.
- **author** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **download_url** [string] A JSON string containing information about the URL of the file.
- **file_url** [string] The URL that may be used to download the file.
- **detected_info** [dict::]

  - **include_header** [boolean] A boolean value indicating whether or not the first row of the file is a header row.

  - **column_delimiter** [string] The column delimiter for the file. One of "comma", "tab", or "pipe".

  - **compression** [string] The type of compression of the file. One of "gzip", or "none".

  - **table_columns** [list::] An array of hashes corresponding to the columns in the file. Each hash should have keys for column "name" and "sql_type" - name : string

      The column name.

      * **sql_type** [string] The SQL type of the column.

get_preprocess_csv(*id*)

  Get a Preprocess CSV

  Parameters

  id [integer]

  Returns

  civis.response.Response

- **id** [integer] The ID of the job created.
- **file_id** [integer] The ID of the file.
- **in_place** [boolean] If true, the file is cleaned in place. If false, a new file ID is created. Defaults to true.
- **detect_table_columns** [boolean] If true, detect the table columns in the file including the sql types. If false, skip table column detection.Defaults to false.
- **force_character_set_conversion** [boolean] If true, the file will always be converted to UTF-8 and any character that cannot be converted will be discarded. If false, the character set conversion will only run if the detected character set is not compatible with UTF-8 (e.g., UTF-8, ASCII).
- **include_header** [boolean] A boolean value indicating whether or not the first row of the file is a header row. If not provided, will attempt to auto-detect whether a header row is present.
- **column_delimiter** [string] The column delimiter for the file. One of "comma", "tab", or "pipe". If not provided, the column delimiter

> > will be auto-detected.
> > - **hidden** [boolean] The hidden status of the item.

list_dependencies(*id*, *, *user_id='DEFAULT'*)

> List dependent objects for this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **user_id** [integer, optional] ID of target user
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **object_type** [string] Dependent object type
> > > > - **fco_type** [string] Human readable dependent object type
> > > > - **id** [integer] Dependent object ID
> > > > - **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > > > - **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > > > - **shareable** [boolean] Whether or not the requesting user can share this object.

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

> List the projects a File belongs to
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the File.
> > >
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **id** [integer] The ID for this project.
> > > > - **author** [dict::]
> > > >
> > > > > - **id** [integer] The ID of this user.
> > > > >
> > > > > - **name** [string] This user's name.
> > > > >
> > > > > - **username** [string] This user's username.
> > > > >
> > > > > - **initials** [string] This user's initials.
> > > > >
> > > > > - **online** [boolean] Whether this user is online.
> > > >
> > > > - **name** [string] The name of this project.
> > > > - **description** [string] A description of the project.
> > > > - **users** [list::] Users who can see the project. - id : integer
> > > >
> > > > > The ID of this user.
> > > >
> > > > > - **name** [string] This user's name.
> > > > >
> > > > > - **username** [string] This user's username.
> > > > >
> > > > > - **initials** [string] This user's initials.
> > > > >
> > > > > - **online** [boolean] Whether this user is online.
> > > >
> > > > - auto_share : boolean
> > > > - created_at : string/time
> > > > - updated_at : string/time
> > > > - **archived** [string] The archival status of the requested item(s).

list_shares(*id*)

List users and groups permissioned on this object

> **Parameters**
>
> > **id** [integer] The ID of the resource that is shared.
>
> **Returns**
>
> > *civis.response.Response*
> >
> > > - **readers** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > > - **writers** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > > - **owners** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch(*id*, ***, *name='DEFAULT'*, *expires_at='DEFAULT'*)

Update details about a file

> **Parameters**
>
> > **id** [integer] The ID of the file.
> >
> > **name** [string, optional] The file name. The extension must match the previous extension.
> >
> > **expires_at** [string/date-time, optional] The date and time the file will expire.
>
> **Returns**
>
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID of the file.
> > > - **name** [string] The file name.
> > > - **created_at** [string/date-time] The date and time the file was created.
> > > - **file_size** [integer] The file size.

- **expires_at** [string/date-time] The date and time the file will expire. If not specified, the file will expire in 30 days. To keep a file indefinitely, specify null.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **download_url** [string] A JSON string containing information about the URL of the file.
- **file_url** [string] The URL that may be used to download the file.
- **detected_info** [dict::]

    - **include_header** [boolean] A boolean value indicating whether or not the first row of the file is a header row.

    - **column_delimiter** [string] The column delimiter for the file. One of "comma", "tab", or "pipe".

    - **compression** [string] The type of compression of the file. One of "gzip", or "none".

    - **table_columns** [list::] An array of hashes corresponding to the columns in the file. Each hash should have keys for column "name" and "sql_type" - name : string

        The column name.

        * **sql_type** [string] The SQL type of the column.

patch_preprocess_csv(*id*, ***, *file_id='DEFAULT'*, *in_place='DEFAULT'*,
            detect_table_columns='DEFAULT', *force_character_set_conversion='DEFAULT'*,
            include_header='DEFAULT', *column_delimiter='DEFAULT'*)

   Update some attributes of this Preprocess CSV

   Parameters

   id [integer] The ID of the job created.

   file_id [integer, optional] The ID of the file.

   in_place [boolean, optional] If true, the file is cleaned in place. If false, a new file ID is created. Defaults to true.

   detect_table_columns [boolean, optional] If true, detect the table columns in the file including the sql types. If false, skip table column detection.Defaults to false.

   force_character_set_conversion [boolean, optional] If true, the file will always be converted to UTF-8 and any character that cannot be converted will be discarded. If false, the character set conversion will only run if the detected character set is not compatible with UTF-8 (e.g., UTF-8, ASCII).

   include_header [boolean, optional] A boolean value indicating whether or not the first row of the file is a header row. If not provided, will attempt to auto-detect whether a header row is present.

   column_delimiter [string, optional] The column delimiter for the file. One of "comma", "tab", or "pipe". If not provided, the column delimiter will be auto-detected.

   Returns

   civis.response.Response

- **id** [integer] The ID of the job created.
- **file_id** [integer] The ID of the file.
- **in_place** [boolean] If true, the file is cleaned in place. If false, a new file
  ID is created. Defaults to true.
- **detect_table_columns** [boolean] If true, detect the table columns in the
  file including the sql types. If false, skip table column detec-
  tion.Defaults to false.
- **force_character_set_conversion** [boolean] If true, the file will always be
  converted to UTF-8 and any character that cannot be converted will
  be discarded. If false, the character set conversion will only run if
  the detected character set is not compatible with UTF-8 (e.g., UTF-8,
  ASCII).
- **include_header** [boolean] A boolean value indicating whether or not the
  first row of the file is a header row. If not provided, will attempt to
  auto-detect whether a header row is present.
- **column_delimiter** [string] The column delimiter for the file. One of
  "comma", "tab", or "pipe". If not provided, the column delimiter
  will be auto-detected.
- **hidden** [boolean] The hidden status of the item.

post(*name*, ***, *expires_at='DEFAULT'*)

Initiate an upload of a file into the platform

> **Parameters**
>
> > **name** [string] The file name.
> >
> > **expires_at** [string/date-time, optional] The date and time the file will expire. If not
> > specified, the file will expire in 30 days. To keep a file indefinitely, specify null.
>
> **Returns**
>
> *civis.response.Response*
>
> > - **id** [integer] The ID of the file.
> > - **name** [string] The file name.
> > - **created_at** [string/date-time] The date and time the file was created.
> > - **file_size** [integer] The file size.
> > - **expires_at** [string/date-time] The date and time the file will expire. If not
> >   specified, the file will expire in 30 days. To keep a file indefinitely,
> >   specify null.
> > - **upload_url** [string] The URL that may be used to upload a file. To use
> >   the upload URL, initiate a POST request to the given URL with the
> >   file you wish to import as the "file" form field.
> > - **upload_fields** [dict] A hash containing the form fields to be included with
> >   the POST request.

post_multipart(*name*, *num_parts*, ***, *expires_at='DEFAULT'*)

Initiate a multipart upload

> **Parameters**
>
> > **name** [string] The file name.
> >
> > **num_parts** [integer] The number of parts in which the file will be uploaded. This
> > parameter determines the number of presigned URLs that are returned.
> >
> > **expires_at** [string/date-time, optional] The date and time the file will expire. If not
> > specified, the file will expire in 30 days. To keep a file indefinitely, specify null.
>
> **Returns**
>
> *civis.response.Response*
>
> > - **id** [integer] The ID of the file.
> > - **name** [string] The file name.
> > - **created_at** [string/date-time] The date and time the file was created.
> > - **file_size** [integer] The file size.

- **expires_at** [string/date-time] The date and time the file will expire. If not specified, the file will expire in 30 days. To keep a file indefinitely, specify null.
- **upload_urls** [list] An array of URLs that may be used to upload file parts. Use separate PUT requests to complete the part uploads. Links expire after 12 hours.

post_multipart_complete(*id*)

Complete a multipart upload

> **Parameters**
>> **id** [integer] The ID of the file.
>
> **Returns**
>> **None** Response code 204: success

post_preprocess_csv(*file_id*, *, *in_place='DEFAULT'*, *detect_table_columns='DEFAULT'*, *force_character_set_conversion='DEFAULT'*, *include_header='DEFAULT'*, *column_delimiter='DEFAULT'*, *hidden='DEFAULT'*)

Create a Preprocess CSV

> **Parameters**
>> **file_id** [integer] The ID of the file.
>>
>> **in_place** [boolean, optional] If true, the file is cleaned in place. If false, a new file ID is created. Defaults to true.
>>
>> **detect_table_columns** [boolean, optional] If true, detect the table columns in the file including the sql types. If false, skip table column detection.Defaults to false.
>>
>> **force_character_set_conversion** [boolean, optional] If true, the file will always be converted to UTF-8 and any character that cannot be converted will be discarded. If false, the character set conversion will only run if the detected character set is not compatible with UTF-8 (e.g., UTF-8, ASCII).
>>
>> **include_header** [boolean, optional] A boolean value indicating whether or not the first row of the file is a header row. If not provided, will attempt to auto-detect whether a header row is present.
>>
>> **column_delimiter** [string, optional] The column delimiter for the file. One of "comma", "tab", or "pipe". If not provided, the column delimiter will be auto-detected.
>>
>> **hidden** [boolean, optional] The hidden status of the item.
>
> **Returns**
>> *civis.response.Response*
>>
>>> - **id** [integer] The ID of the job created.
>>> - **file_id** [integer] The ID of the file.
>>> - **in_place** [boolean] If true, the file is cleaned in place. If false, a new file ID is created. Defaults to true.
>>> - **detect_table_columns** [boolean] If true, detect the table columns in the file including the sql types. If false, skip table column detection.Defaults to false.
>>> - **force_character_set_conversion** [boolean] If true, the file will always be converted to UTF-8 and any character that cannot be converted will be discarded. If false, the character set conversion will only run if the detected character set is not compatible with UTF-8 (e.g., UTF-8, ASCII).
>>> - **include_header** [boolean] A boolean value indicating whether or not the first row of the file is a header row. If not provided, will attempt to auto-detect whether a header row is present.
>>> - **column_delimiter** [string] The column delimiter for the file. One of "comma", "tab", or "pipe". If not provided, the column delimiter will be auto-detected.

> • **hidden** [boolean] The hidden status of the item.

put(*id*, *name*, *expires_at*)

    Update details about a file

        Parameters

            id [integer] The ID of the file.

            name [string] The file name. The extension must match the previous extension.

            expires_at [string/date-time] The date and time the file will expire.

        Returns

            civis.response.Response

                • **id** [integer] The ID of the file.

                • **name** [string] The file name.

                • **created_at** [string/date-time] The date and time the file was created.

                • **file_size** [integer] The file size.

                • **expires_at** [string/date-time] The date and time the file will expire. If not specified, the file will expire in 30 days. To keep a file indefinitely, specify null.

                • **author** [dict::]

                    – **id** [integer] The ID of this user.

                    – **name** [string] This user's name.

                    – **username** [string] This user's username.

                    – **initials** [string] This user's initials.

                    – **online** [boolean] Whether this user is online.

                • **download_url** [string] A JSON string containing information about the URL of the file.

                • **file_url** [string] The URL that may be used to download the file.

                • **detected_info** [dict::]

                    – **include_header** [boolean] A boolean value indicating whether or not the first row of the file is a header row.

                    – **column_delimiter** [string] The column delimiter for the file. One of "comma", "tab", or "pipe".

                    – **compression** [string] The type of compression of the file. One of "gzip", or "none".

                    – **table_columns** [list::] An array of hashes corresponding to the columns in the file. Each hash should have keys for column "name" and "sql_type" - name : string

                        The column name.

                        ∗ **sql_type** [string] The SQL type of the column.

put_preprocess_csv(*id*, *file_id*, ***, *in_place='DEFAULT'*, *detect_table_columns='DEFAULT'*, *force_character_set_conversion='DEFAULT'*, *include_header='DEFAULT'*, *column_delimiter='DEFAULT'*)

    Replace all attributes of this Preprocess CSV

        Parameters

            id [integer] The ID of the job created.

            file_id [integer] The ID of the file.

            in_place [boolean, optional] If true, the file is cleaned in place. If false, a new file ID is created. Defaults to true.

detect_table_columns [boolean, optional] If true, detect the table columns in the file including the sql types. If false, skip table column detection.Defaults to false.

force_character_set_conversion [boolean, optional] If true, the file will always be converted to UTF-8 and any character that cannot be converted will be discarded. If false, the character set conversion will only run if the detected character set is not compatible with UTF-8 (e.g., UTF-8, ASCII).

include_header [boolean, optional] A boolean value indicating whether or not the first row of the file is a header row. If not provided, will attempt to auto-detect whether a header row is present.

column_delimiter [string, optional] The column delimiter for the file. One of "comma", "tab", or "pipe". If not provided, the column delimiter will be auto-detected.

Returns

civis.response.Response

- **id** [integer] The ID of the job created.
- **file_id** [integer] The ID of the file.
- **in_place** [boolean] If true, the file is cleaned in place. If false, a new file ID is created. Defaults to true.
- **detect_table_columns** [boolean] If true, detect the table columns in the file including the sql types. If false, skip table column detection.Defaults to false.
- **force_character_set_conversion** [boolean] If true, the file will always be converted to UTF-8 and any character that cannot be converted will be discarded. If false, the character set conversion will only run if the detected character set is not compatible with UTF-8 (e.g., UTF-8, ASCII).
- **include_header** [boolean] A boolean value indicating whether or not the first row of the file is a header row. If not provided, will attempt to auto-detect whether a header row is present.
- **column_delimiter** [string] The column delimiter for the file. One of "comma", "tab", or "pipe". If not provided, the column delimiter will be auto-detected.
- **hidden** [boolean] The hidden status of the item.

put_preprocess_csv_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

civis.response.Response

- **id** [integer] The ID of the job created.
- **file_id** [integer] The ID of the file.
- **in_place** [boolean] If true, the file is cleaned in place. If false, a new file ID is created. Defaults to true.
- **detect_table_columns** [boolean] If true, detect the table columns in the file including the sql types. If false, skip table column detection.Defaults to false.
- **force_character_set_conversion** [boolean] If true, the file will always be converted to UTF-8 and any character that cannot be converted will be discarded. If false, the character set conversion will only run if the detected character set is not compatible with UTF-8 (e.g., UTF-8, ASCII).
- **include_header** [boolean] A boolean value indicating whether or not the

> > first row of the file is a header row. If not provided, will attempt to
> > auto-detect whether a header row is present.
>
> - **column_delimiter** [string] The column delimiter for the file. One of
>   "comma", "tab", or "pipe". If not provided, the column delimiter
>   will be auto-detected.
> - **hidden** [boolean] The hidden status of the item.

put_projects(*id*, *project_id*)

> Add a File to a project
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the File.
> > >
> > > **project_id** [integer] The ID of the project.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

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

> Set the permissions groups has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **group_ids** [list] An array of one or more group IDs.
> > >
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > >
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > >
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > [*civis.response.Response*](#)
> >
> > > - **readers** [dict::]
> > >
> > >   > - **users** [list::]
> > >   >
> > >   >   > * id : integer
> > >   >   >
> > >   >   > * name : string
> > >   >
> > >   > - **groups** [list::]
> > >   >
> > >   >   > * id : integer
> > >   >   >
> > >   >   > * name : string
> > >
> > > - **writers** [dict::]
> > >
> > >   > - **users** [list::]
> > >   >
> > >   >   > * id : integer
> > >   >   >
> > >   >   > * name : string
> > >   >
> > >   > - **groups** [list::]
> > >   >
> > >   >   > * id : integer
> > >   >   >
> > >   >   > * name : string
> > >
> > > - **owners** [dict::]
> > >
> > >   > - **users** [list::]
> > >   >
> > >   >   > * id : integer
> > >   >   >
> > >   >   > * name : string
> > >   >
> > >   > - **groups** [list::]
> > >   >
> > >   >   > * id : integer

               * name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

    Set the permissions users have on this object

        Parameters

            id [integer] The ID of the resource that is shared.

            user_ids [list] An array of one or more user IDs.

            permission_level [string] Options are: "read", "write", or "manage".

            share_email_body [string, optional] Custom body text for e-mail sent on a share.

            send_shared_email [boolean, optional] Send email to the recipients of a share.

        Returns

        civis.response.Response

- **readers** [dict::]

        – **users** [list::]

            * id : integer

            * name : string

        – **groups** [list::]

            * id : integer

            * name : string

- **writers** [dict::]

        – **users** [list::]

            * id : integer

            * name : string

        – **groups** [list::]

            * id : integer

            * name : string

- **owners** [dict::]

        – **users** [list::]

            * id : integer

            * name : string

         – **groups** [list::]

            * id : integer

            * name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_transfer(*id*, *user_id*, *include_dependencies*, *, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

    Transfer ownership of this object to another user

        Parameters

> **id** [integer] The ID of the resource that is shared.
>
> **user_id** [integer] ID of target user
>
> **include_dependencies** [boolean] Whether or not to give manage permissions on all dependencies
>
> **email_body** [string, optional] Custom body text for e-mail sent on transfer.
>
> **send_email** [boolean, optional] Send email to the target user of the transfer?

Returns

> *civis.response.Response*
>
> > • **dependencies** [list::] Dependent objects for this object - object_type : string
> >
> > > Dependent object type
> >
> > > – **fco_type** [string] Human readable dependent object type
> > >
> > > – **id** [integer] Dependent object ID
> > >
> > > – **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > >
> > > – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > >
> > > – **shared** [boolean] Whether dependent object was successfully shared with target user

Git_Repos

class Git_Repos(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.git_repos.list(...)
```

Methods

delete(id)	Remove the bookmark on a git repository
get(id)	Get a bookmarked git repository
list(*[, limit, page_num, order, order_dir, ...])	List bookmarked git repositories
list_refs(id)	Get all branches and tags of a bookmarked git repository
post(repo_url)	Bookmark a git repository

———

delete(*id*)

> Remove the bookmark on a git repository
>
> > **Parameters**

> > > **id** [integer] The ID for this git repository.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

> **get**(*id*)
>
> > Get a bookmarked git repository
> >
> > > **Parameters**
> > >
> > > > **id** [integer] The ID for this git repository.
> > >
> > > **Returns**
> > >
> > > > [*civis.response.Response*](#)
> > > >
> > > > > - **id** [integer] The ID for this git repository.
> > > > > - **repo_url** [string] The URL for this git repository.
> > > > > - created_at : string/time
> > > > > - updated_at : string/time

> **list**(***, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)
>
> > List bookmarked git repositories
> >
> > > **Parameters**
> > >
> > > > **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 repo_url. Must be one of: repo_url, 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**
> > >
> > > > [*civis.response.PaginatedResponse*](#)
> > > >
> > > > > - **id** [integer] The ID for this git repository.
> > > > > - **repo_url** [string] The URL for this git repository.
> > > > > - created_at : string/time
> > > > > - updated_at : string/time

> **list_refs**(*id*)
>
> > Get all branches and tags of a bookmarked git repository
> >
> > > **Parameters**
> > >
> > > > **id** [integer] The ID for this git repository.
> > >
> > > **Returns**
> > >
> > > > [*civis.response.Response*](#)
> > > >
> > > > > - **branches** [list] List of branch names of this git repository.
> > > > > - **tags** [list] List of tag names of this git repository.

> **post**(*repo_url*)
>
> > Bookmark a git repository
> >
> > > **Parameters**
> > >
> > > > **repo_url** [string] The URL for this git repository.
> > >
> > > **Returns**
> > >
> > > > [*civis.response.Response*](#)
> > > >
> > > > > - **id** [integer] The ID for this git repository.
> > > > > - **repo_url** [string] The URL for this git repository.
> > > > > - created_at : string/time
> > > > > - updated_at : string/time

Groups

class Groups(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.groups.list(...)
```

Methods

delete_members(id, user_id)	Remove a user from a group
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 a Group
list(*[, query, permission, ...])	List Groups
list_child_groups(id)	Get child groups of this group
list_shares(id)	List users and groups permissioned on this object
patch(id, *[, name, description, slug, ...])	Update some attributes of this Group
post(name, *[, description, slug, ...])	Create a Group
put(id, name, *[, description, slug, ...])	Replace all attributes of this Group
put_members(id, user_id)	Add a user to a group
put_shares_groups(id, group_ids, ...[, ...])	Set the permissions groups has on this object
put_shares_users(id, user_ids, ...[, ...])	Set the permissions users have on this object

—

> **delete_members**(*id*, *user_id*)
>> Remove a user from a group
>>> **Parameters**
>>>> **id** [integer] The ID of the group.
>>>> **user_id** [integer] The ID of the user.
>>> **Returns**
>>>> **None** Response code 204: success

> **delete_shares_groups**(*id*, *group_id*)
>> Revoke the permissions a group has on this object
>>> **Parameters**
>>>> **id** [integer] The ID of the resource that is shared.
>>>> **group_id** [integer] The 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] The ID of the resource that is shared.
>>>> **user_id** [integer] The ID of the user.
>>> **Returns**

> > > > **None** Response code 204: success

get(*id*)
> > Get a Group
> > > **Parameters**
> > > > **id** [integer]
> > > **Returns**

> > > > *civis.response.Response*
> > > > > > • **id** [integer] The ID of this group.
> > > > > > • **name** [string] This group's name.
> > > > > > • **created_at** [string/time] The date and time when this group was created.
> > > > > > • **updated_at** [string/time] The date and time when this group was last updated.
> > > > > > • **description** [string] The description of the group.
> > > > > > • **slug** [string] The slug for this group.
> > > > > > • **organization_id** [integer] The ID of the organization this group belongs to.
> > > > > > • **organization_name** [string] The name of the organization this group belongs to.
> > > > > > • **member_count** [integer] The number of active members in this group.
> > > > > > • **total_member_count** [integer] The total number of members in this group.
> > > > > > • **must_agree_to_eula** [boolean] Whether or not members of this group must sign the EULA. Deprecated: all users must agree to the EULA, regardless of this attribute.
> > > > > > • **default_otp_required_for_login** [boolean] The two factor authentication requirement for this group.
> > > > > > • **role_ids** [list] An array of ids of all the roles this group has.
> > > > > > • **default_time_zone** [string] The default time zone of this group.
> > > > > > • **default_jobs_label** [string] The default partition label for jobs of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
> > > > > > • **default_notebooks_label** [string] The default partition label for notebooks of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
> > > > > > • **default_services_label** [string] The default partition label for services of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
> > > > > > • **last_updated_by_id** [integer] The ID of the user who last updated this group.
> > > > > > • **created_by_id** [integer] The ID of the user who created this group.
> > > > > > • **members** [list::] The members of this group. - id : integer

> > > > > > > The ID of this user.

> > > > > > > – **name** [string] This user's name.

> > > > > > > – **username** [string] This user's username.

> > > > > > > – **initials** [string] This user's initials.

> > > > > > > – **online** [boolean] Whether this user is online.

> > > > > > > – **email** [string] This user's email address.

> > > > > > > – **primary_group_id** [integer] The ID of the primary group of this user.

– **active** [boolean] The account status of this user.

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

> List Groups
>
> > **Parameters**
> >
> > > **query** [string, optional] If specified, it will filter the groups returned. Infix matching is supported (e.g., "query=group" will return "group" and "group of people" and "my group" and "my group of people").
> > >
> > > **permission** [string, optional] A permissions string, one of "read", "write", or "manage". Lists only groups for which the current user has that permission.
> > >
> > > **include_members** [boolean, optional] Show members of the group.
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to 50. Maximum allowed is 1000.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to name. Must be one of: name, 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**
> >
> > > *civis.response.PaginatedResponse*
> > >
> > > > - **id** [integer] The ID of this group.
> > > > - **name** [string] This group's name.
> > > > - **created_at** [string/time] The date and time when this group was created.
> > > > - **updated_at** [string/time] The date and time when this group was last updated.
> > > > - **slug** [string] The slug for this group.
> > > > - **organization_id** [integer] The ID of the organization this group belongs to.
> > > > - **organization_name** [string] The name of the organization this group belongs to.
> > > > - **member_count** [integer] The number of active members in this group.
> > > > - **total_member_count** [integer] The total number of members in this group.
> > > > - **last_updated_by_id** [integer] The ID of the user who last updated this group.
> > > > - **created_by_id** [integer] The ID of the user who created this group.
> > > > - **members** [list::] The members of this group. - id : integer
> > > >
> > > > > The ID of this user.
> > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.

list_child_groups(*id*)

> Get child groups of this group
>
> > **Parameters**
> >
> > > **id** [integer] The ID of this group.

>>> **Returns**
>>> [*civis.response.Response*](#)
>>>> • **manageable** [list::]
>>>>> – id : integer
>>>>> – name : string
>>>> • **writeable** [list::]
>>>>> – id : integer
>>>>> – name : string
>>>> • **readable** [list::]
>>>>> – id : integer
>>>>> – name : string

> **list_shares**(*id*)
>> List users and groups permissioned on this object
>>> **Parameters**
>>>> **id** [integer] The ID of the resource that is shared.
>>> **Returns**
>>> [*civis.response.Response*](#)
>>>> • **readers** [dict::]
>>>>> – **users** [list::]
>>>>>> ∗ id : integer
>>>>>> ∗ name : string
>>>>> – **groups** [list::]
>>>>>> ∗ id : integer
>>>>>> ∗ name : string
>>>> • **writers** [dict::]
>>>>> – **users** [list::]
>>>>>> ∗ id : integer
>>>>>> ∗ name : string
>>>>> – **groups** [list::]
>>>>>> ∗ id : integer
>>>>>> ∗ name : string
>>>> • **owners** [dict::]
>>>>> – **users** [list::]
>>>>>> ∗ id : integer
>>>>>> ∗ name : string
>>>>> – **groups** [list::]
>>>>>> ∗ id : integer
>>>>>> ∗ name : string
>>>> • **total_user_shares** [integer] For owners, the number of total users shared.
>>>> For writers and readers, the number of visible users shared.

> > > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch(*id*, ***, *name='DEFAULT'*, *description='DEFAULT'*, *slug='DEFAULT'*, *organization_id='DEFAULT'*, *must_agree_to_eula='DEFAULT'*, *default_otp_required_for_login='DEFAULT'*, *role_ids='DEFAULT'*, *default_time_zone='DEFAULT'*, *default_jobs_label='DEFAULT'*, *default_notebooks_label='DEFAULT'*, *default_services_label='DEFAULT'*)

> Update some attributes of this Group
>
> > **Parameters**
> >
> > > **id** [integer] The ID of this group.
> > > **name** [string, optional] This group's name.
> > > **description** [string, optional] The description of the group.
> > > **slug** [string, optional] The slug for this group.
> > > **organization_id** [integer, optional] The ID of the organization this group belongs to.
> > > **must_agree_to_eula** [boolean, optional] Whether or not members of this group must sign the EULA. Deprecated: all users must agree to the EULA, regardless of this attribute.
> > > **default_otp_required_for_login** [boolean, optional] The two factor authentication requirement for this group.
> > > **role_ids** [list, optional] An array of ids of all the roles this group has.
> > > **default_time_zone** [string, optional] The default time zone of this group.
> > > **default_jobs_label** [string, optional] The default partition label for jobs of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
> > > **default_notebooks_label** [string, optional] The default partition label for notebooks of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
> > > **default_services_label** [string, optional] The default partition label for services of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **id** [integer] The ID of this group.
> > > > - **name** [string] This group's name.
> > > > - **created_at** [string/time] The date and time when this group was created.
> > > > - **updated_at** [string/time] The date and time when this group was last updated.
> > > > - **description** [string] The description of the group.
> > > > - **slug** [string] The slug for this group.
> > > > - **organization_id** [integer] The ID of the organization this group belongs to.
> > > > - **organization_name** [string] The name of the organization this group belongs to.
> > > > - **member_count** [integer] The number of active members in this group.
> > > > - **total_member_count** [integer] The total number of members in this group.
> > > > - **must_agree_to_eula** [boolean] Whether or not members of this group must sign the EULA. Deprecated: all users must agree to the EULA, regardless of this attribute.
> > > > - **default_otp_required_for_login** [boolean] The two factor authentication requirement for this group.
> > > > - **role_ids** [list] An array of ids of all the roles this group has.
> > > > - **default_time_zone** [string] The default time zone of this group.
> > > > - **default_jobs_label** [string] The default partition label for jobs of this

group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.

- **default_notebooks_label** [string] The default partition label for notebooks of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
- **default_services_label** [string] The default partition label for services of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
- **last_updated_by_id** [integer] The ID of the user who last updated this group.
- **created_by_id** [integer] The ID of the user who created this group.
- **members** [list::] The members of this group. - id : integer

    The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

    – **email** [string] This user's email address.

    – **primary_group_id** [integer] The ID of the primary group of this user.

    – **active** [boolean] The account status of this user.

post(*name*, ***, *description='DEFAULT'*, *slug='DEFAULT'*, *organization_id='DEFAULT'*, *must_agree_to_eula='DEFAULT'*, *default_otp_required_for_login='DEFAULT'*, *role_ids='DEFAULT'*, *default_time_zone='DEFAULT'*, *default_jobs_label='DEFAULT'*, *default_notebooks_label='DEFAULT'*, *default_services_label='DEFAULT'*)

Create a Group

> **Parameters**
>
> > **name** [string] This group's name.
> >
> > **description** [string, optional] The description of the group.
> >
> > **slug** [string, optional] The slug for this group.
> >
> > **organization_id** [integer, optional] The ID of the organization this group belongs to.
> >
> > **must_agree_to_eula** [boolean, optional] Whether or not members of this group must sign the EULA. Deprecated: all users must agree to the EULA, regardless of this attribute.
> >
> > **default_otp_required_for_login** [boolean, optional] The two factor authentication requirement for this group.
> >
> > **role_ids** [list, optional] An array of ids of all the roles this group has.
> >
> > **default_time_zone** [string, optional] The default time zone of this group.
> >
> > **default_jobs_label** [string, optional] The default partition label for jobs of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
> >
> > **default_notebooks_label** [string, optional] The default partition label for notebooks of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
> >
> > **default_services_label** [string, optional] The default partition label for services of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
>
> **Returns**
>
> > *civis.response.Response*

- **id** [integer] The ID of this group.
- **name** [string] This group's name.
- **created_at** [string/time] The date and time when this group was created.
- **updated_at** [string/time] The date and time when this group was last updated.
- **description** [string] The description of the group.
- **slug** [string] The slug for this group.
- **organization_id** [integer] The ID of the organization this group belongs to.
- **organization_name** [string] The name of the organization this group belongs to.
- **member_count** [integer] The number of active members in this group.
- **total_member_count** [integer] The total number of members in this group.
- **must_agree_to_eula** [boolean] Whether or not members of this group must sign the EULA. Deprecated: all users must agree to the EULA, regardless of this attribute.
- **default_otp_required_for_login** [boolean] The two factor authentication requirement for this group.
- **role_ids** [list] An array of ids of all the roles this group has.
- **default_time_zone** [string] The default time zone of this group.
- **default_jobs_label** [string] The default partition label for jobs of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
- **default_notebooks_label** [string] The default partition label for notebooks of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
- **default_services_label** [string] The default partition label for services of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
- **last_updated_by_id** [integer] The ID of the user who last updated this group.
- **created_by_id** [integer] The ID of the user who created this group.
- **members** [list::] The members of this group. - id : integer

    The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

    – **email** [string] This user's email address.

    – **primary_group_id** [integer] The ID of the primary group of this user.

    – **active** [boolean] The account status of this user.

put(*id*, *name*, ***, *description='DEFAULT'*, *slug='DEFAULT'*, *organization_id='DEFAULT'*, *must_agree_to_eula='DEFAULT'*, *default_otp_required_for_login='DEFAULT'*, *role_ids='DEFAULT'*, *default_time_zone='DEFAULT'*, *default_jobs_label='DEFAULT'*, *default_notebooks_label='DEFAULT'*, *default_services_label='DEFAULT'*)

Replace all attributes of this Group

Parameters

id [integer] The ID of this group.

name [string] This group's name.

description [string, optional] The description of the group.

slug [string, optional] The slug for this group.

organization_id [integer, optional] The ID of the organization this group belongs to.

must_agree_to_eula [boolean, optional] Whether or not members of this group must sign the EULA. Deprecated: all users must agree to the EULA, regardless of this attribute.

default_otp_required_for_login [boolean, optional] The two factor authentication requirement for this group.

role_ids [list, optional] An array of ids of all the roles this group has.

default_time_zone [string, optional] The default time zone of this group.

default_jobs_label [string, optional] The default partition label for jobs of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.

default_notebooks_label [string, optional] The default partition label for notebooks of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.

default_services_label [string, optional] The default partition label for services of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.

Returns

civis.response.Response

- **id** [integer] The ID of this group.
- **name** [string] This group's name.
- **created_at** [string/time] The date and time when this group was created.
- **updated_at** [string/time] The date and time when this group was last updated.
- **description** [string] The description of the group.
- **slug** [string] The slug for this group.
- **organization_id** [integer] The ID of the organization this group belongs to.
- **organization_name** [string] The name of the organization this group belongs to.
- **member_count** [integer] The number of active members in this group.
- **total_member_count** [integer] The total number of members in this group.
- **must_agree_to_eula** [boolean] Whether or not members of this group must sign the EULA. Deprecated: all users must agree to the EULA, regardless of this attribute.
- **default_otp_required_for_login** [boolean] The two factor authentication requirement for this group.
- **role_ids** [list] An array of ids of all the roles this group has.
- **default_time_zone** [string] The default time zone of this group.
- **default_jobs_label** [string] The default partition label for jobs of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
- **default_notebooks_label** [string] The default partition label for notebooks of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
- **default_services_label** [string] The default partition label for services of this group. Only available if custom_partitions feature flag is set. Do not use this attribute as it may break in the future.
- **last_updated_by_id** [integer] The ID of the user who last updated this

group.
- **created_by_id** [integer] The ID of the user who created this group.
- **members** [list::] The members of this group. - id : integer

  The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

    – **email** [string] This user's email address.

    – **primary_group_id** [integer] The ID of the primary group of
      this user.

    – **active** [boolean] The account status of this user.

put_members(*id*, *user_id*)
    Add a user to a group
        Parameters
            id [integer] The ID of the group.
            user_id [integer] The ID of the user.
        Returns
        civis.response.Response
- **id** [integer] The ID of this group.
- **name** [string] This group's name.
- **created_at** [string/time] The date and time when this group was created.
- **updated_at** [string/time] The date and time when this group was last up-
  dated.
- **description** [string] The description of the group.
- **slug** [string] The slug for this group.
- **organization_id** [integer] The ID of the organization this group belongs
  to.
- **organization_name** [string] The name of the organization this group be-
  longs to.
- **member_count** [integer] The number of active members in this group.
- **total_member_count** [integer] The total number of members in this
  group.
- **must_agree_to_eula** [boolean] Whether or not members of this group
  must sign the EULA. Deprecated: all users must agree to the EULA,
  regardless of this attribute.
- **default_otp_required_for_login** [boolean] The two factor authentication
  requirement for this group.
- **role_ids** [list] An array of ids of all the roles this group has.
- **default_time_zone** [string] The default time zone of this group.
- **default_jobs_label** [string] The default partition label for jobs of this
  group. Only available if custom_partitions feature flag is set. Do
  not use this attribute as it may break in the future.
- **default_notebooks_label** [string] The default partition label for note-
  books of this group. Only available if custom_partitions feature flag
  is set. Do not use this attribute as it may break in the future.
- **default_services_label** [string] The default partition label for services of
  this group. Only available if custom_partitions feature flag is set.
  Do not use this attribute as it may break in the future.

- **last_updated_by_id** [integer] The ID of the user who last updated this group.
- **created_by_id** [integer] The ID of the user who created this group.
- **members** [list::] The members of this group. - id : integer

> The ID of this user.

>> – **name** [string] This user's name.
>>
>> – **username** [string] This user's username.
>>
>> – **initials** [string] This user's initials.
>>
>> – **online** [boolean] Whether this user is online.
>>
>> – **email** [string] This user's email address.
>>
>> – **primary_group_id** [integer] The ID of the primary group of this user.
>>
>> – **active** [boolean] The account status of this user.

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

> Set the permissions groups has on this object
>
> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>> **group_ids** [list] An array of one or more group IDs.
>> **permission_level** [string] Options are: "read", "write", or "manage".
>> **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
>> **send_shared_email** [boolean, optional] Send email to the recipients of a share.
>
> **Returns**
>> [*civis.response.Response*](#)
>>> - **readers** [dict::]
>>>
>>>> – **users** [list::]
>>>>
>>>>> * id : integer
>>>>>
>>>>> * name : string
>>>>
>>>> – **groups** [list::]
>>>>
>>>>> * id : integer
>>>>>
>>>>> * name : string
>>>
>>> - **writers** [dict::]
>>>
>>>> – **users** [list::]
>>>>
>>>>> * id : integer
>>>>>
>>>>> * name : string
>>>>
>>>> – **groups** [list::]
>>>>
>>>>> * id : integer
>>>>>
>>>>> * name : string
>>>
>>> - **owners** [dict::]
>>>
>>>> – **users** [list::]
>>>>
>>>>> * id : integer

         ∗ name : string

       – groups [list::]

         ∗ id : integer

         ∗ name : string

    • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

    • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

 Set the permissions users have on this object

  Parameters

   id [integer] The ID of the resource that is shared.

   user_ids [list] An array of one or more user IDs.

   permission_level [string] Options are: "read", "write", or "manage".

   share_email_body [string, optional] Custom body text for e-mail sent on a share.

   send_shared_email [boolean, optional] Send email to the recipients of a share.

  Returns

  civis.response.Response

    • **readers** [dict::]

      – users [list::]

         ∗ id : integer

         ∗ name : string

      – groups [list::]

         ∗ id : integer

         ∗ name : string

    • **writers** [dict::]

      – users [list::]

         ∗ id : integer

         ∗ name : string

      – groups [list::]

         ∗ id : integer

         ∗ name : string

    • **owners** [dict::]

      – users [list::]

         ∗ id : integer

         ∗ name : string

      – groups [list::]

         ∗ id : integer

         ∗ name : string

    • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

> • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Imports

class **Imports**(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.imports.list_shares(...)
```

Methods

delete_files_csv_runs(id, run_id)	Cancel a run
delete_files_runs(id, run_id)	Cancel a run
delete_projects(id, project_id)	Remove an 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
get(id)	Get details about an import
get_batches(id)	Get details about a batch import
get_files_csv(id)	Get a CSV Import
get_files_csv_runs(id, run_id)	Check status of a run
get_files_runs(id, run_id)	Check status of a run
list(*[, type, destination, source, status, ...])	List Imports
list_batches(*[, hidden, limit, page_num, ...])	List batch imports
list_dependencies(id, *[, user_id])	List dependent objects for this object
list_files_csv_runs(id, *[, limit, ...])	List runs for the given csv_import
list_files_csv_runs_logs(id, run_id, *[, ...])	Get the logs for a run
list_files_runs(id, *[, limit, page_num, ...])	List runs for the given import
list_files_runs_logs(id, run_id, *[, ...])	Get the logs for a run
list_projects(id, *[, hidden])	List the projects an Import belongs to
list_runs(id)	Get the run history of this import
list_runs_logs(id, run_id, *[, last_id, limit])	Get the logs for a run
list_shares(id)	List users and groups permissioned on this object
patch_files_csv(id, *[, name, source, ...])	Update some attributes of this CSV Import
post(name, sync_type, is_outbound, *[, ...])	Create a new import configuration
post_batches(file_ids, schema, table, ...[, ...])	Upload multiple files to Civis
post_cancel(id)	Cancel a run
post_files(schema, name, remote_host_id, ...)	Initate an import of a tabular file into the platform
post_files_csv(source, destination, ...[, ...])	Create a CSV Import
post_files_csv_runs(id)	Start a run
post_files_runs(id)	Start a run
post_runs(id)	Run an import
post_syncs(id, source, destination, *[, ...])	Create a sync
put(id, name, sync_type, is_outbound, *[, ...])	Update an import

Table 31 – continued from previous page

put_archive(id, status)	Update the archive status of this object
put_files_csv(id, source, destination, ...)	Replace all attributes of this CSV Import
put_files_csv_archive(id, status)	Update the archive status of this object
put_projects(id, project_id)	Add an Import to a project
put_shares_groups(id, group_ids, ...[, ...])	Set the permissions groups has on this object
put_shares_users(id, user_ids, ...[, ...])	Set the permissions users have on this object
put_syncs(id, sync_id, source, destination, *)	Update a sync
put_syncs_archive(id, sync_id, *[, status])	Update the archive status of this sync
put_transfer(id, user_id, ...[, email_body, ...])	Transfer ownership of this object to another user

——

delete_files_csv_runs(*id*, *run_id*)

> Cancel a run
> > **Parameters**
> > > **id** [integer] The ID of the csv_import.
> > > **run_id** [integer] The ID of the run.
> > **Returns**
> > > **None** Response code 202: success

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 an Import from a project
> > **Parameters**
> > > **id** [integer] The ID of the Import.
> > > **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] The ID of the resource that is shared.
> > > **group_id** [integer] The 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] The ID of the resource that is shared.
> > > **user_id** [integer] The ID of the user.
> > **Returns**
> > > **None** Response code 204: success

get(*id*)

> Get details about an import
> > **Parameters**

id [integer] The ID for the import.

Returns

[*civis.response.Response*](#)

- **name** [string] The name of the import.

- **sync_type** [string] The type of sync to perform; one of Dbsync, AutoImport, GdocImport, GdocExport, and Salesforce.

- **source** [dict::]

  – remote_host_id : integer

  – credential_id : integer

  – **additional_credentials** [list] Array that holds additional credentials used for specific imports. For salesforce imports, the first and only element is the client credential id. For DB Syncs, the first element is an SSL private key credential id, and the second element is the corresponding public key credential id.

  – name : string

- **destination** [dict::]

  – remote_host_id : integer

  – credential_id : integer

  – **additional_credentials** [list] Array that holds additional credentials used for specific imports. For salesforce imports, the first and only element is the client credential id. For DB Syncs, the first element is an SSL private key credential id, and the second element is the corresponding public key credential id.

  – name : string

- **schedule** [dict::]

  – **scheduled** [boolean] If the item is scheduled.

  – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  – **scheduled_hours** [list] Hours of the day it is scheduled on.

  – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

  – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

  – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

  – **urls** [list] URLs to receive a POST request at job completion

  – **success_email_subject** [string] Custom subject line for success e-mail.

  – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

- **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

- **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

- **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.

- **parent_id** [integer] Parent id to trigger this import from
- **id** [integer] The ID for the import.
- is_outbound : boolean
- **job_type** [string] The job type of this import.
- **syncs** [list::] List of syncs. - id : integer - source : dict:

```
- id : integer
    The ID of the table or file, if available.
- path : string
    The path of the dataset to sync from; for a␣
→database source,
    schema.tablename. If you are doing a Google␣
→Sheet export, this
    can be blank. This is a legacy parameter, it␣
→is recommended you
    use one of the following: databaseTable, file,␣
→googleWorksheet,
    salesforce
- database_table : dict::
    - schema : string
        The database schema name.
    - table : string
        The database table name.
    - use_without_schema : boolean
        This attribute is no longer available;␣
→defaults to false
        but cannot be used.
- file : dict::
    - id : integer
        The file id.
- google_worksheet : dict::
    - spreadsheet : string
        The spreadsheet document name.
    - spreadsheet_id : string
        The spreadsheet document id.
    - worksheet : string
        The worksheet tab name.
    - worksheet_id : integer
```

```
        The worksheet tab id.
- salesforce : dict::
    - object_name : string
        The Salesforce object name.
```

- **destination** [dict::]

  * **path** [string] The schema.tablename to sync to. If you are doing a Google Sheet export, this is the spreadsheet and sheet name separated by a period. i.e. if you have a spreadsheet named "MySpreadsheet" and a sheet called "Sheet1" this field would be "MySpreadsheet.Sheet1". This is a legacy parameter, it is recommended you use one of the following: databaseTable, googleWorksheet

  * **database_table** [dict::]

    · **schema** [string] The database schema name.

    · **table** [string] The database table name.

    · **use_without_schema** [boolean] This attribute is no longer available; defaults to false but cannot be used.

  * **google_worksheet** [dict::]

    · **spreadsheet** [string] The spreadsheet document name.

    · **spreadsheet_id** [string] The spreadsheet document id.

    · **worksheet** [string] The worksheet tab name.

    · **worksheet_id** [integer] The worksheet tab id.

- **advanced_options** [dict::]

  * max_errors : integer

  * existing_table_rows : string

  * diststyle : string

  * distkey : string

  * sortkey1 : string

  * sortkey2 : string

  * column_delimiter : string

  * **column_overrides** [dict] Hash used for overriding auto-detected names and types, with keys being the index of the column being overridden.

* **escaped** [boolean] If true, escape quotes with a backslash; otherwise, escape quotes by double-quoting. Defaults to false.

* identity_column : string

* row_chunk_size : integer

* wipe_destination_table : boolean

* truncate_long_lines : boolean

* invalid_char_replacement : string

* verify_table_row_counts : boolean

* **partition_column_name** [string] This parameter is deprecated

* **partition_schema_name** [string] This parameter is deprecated

* **partition_table_name** [string] This parameter is deprecated

* **partition_table_partition_column_min_name** [string] This parameter is deprecated

* **partition_table_partition_column_max_name** [string] This parameter is deprecated

* last_modified_column : string

* **mysql_catalog_matches_schema** [boolean] This attribute is no longer available; defaults to true but cannot be used.

* **chunking_method** [string] This parameter is deprecated

* first_row_is_header : boolean

* **export_action** [string] The kind of export action you want to have the export execute. Set to "newsprsht" if you want a new worksheet inside a new spreadsheet. Set to "newwksht" if you want a new worksheet inside an existing spreadsheet. Set to "updatewksht" if you want to overwrite an existing worksheet inside an existing spreadsheet. Set to "appendwksht" if you want to append to the end of an existing worksheet inside an existing spreadsheet. Default is set to "newsprsht"

* **sql_query** [string] If you are doing a Google Sheet export, this is your SQL query.

* contact_lists : string

* soql_query : string

* include_deleted_records : boolean

• state : string

- created_at : string/date-time
- updated_at : string/date-time
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **user** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this import.
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

get_batches(*id*)

    Get details about a batch import

    Parameters

    id [integer] The ID for the import.

    Returns

    civis.response.Response

- **id** [integer] The ID for the import.
- **schema** [string] The destination schema name. This schema must already exist in Redshift.
- **table** [string] The destination table name, without the schema prefix. This table must already exist in Redshift.
- **remote_host_id** [integer] The ID of the destination database host.
- **state** [string] The state of the run; one of "queued", "running", "succeeded", "failed", or "cancelled".
- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.
- **error** [string] The error returned by the run, if any.
- **hidden** [boolean] The hidden status of the item.

get_files_csv(*id*)

Get a CSV Import

> **Parameters**
>
> > **id** [integer]
>
> **Returns**
>
> > *civis.response.Response*
> >
> > - **id** [integer] The ID for the import.
> > - **name** [string] The name of the import.
> > - **source** [dict::]
> >
> >   - **file_ids** [list] The file ID(s) to import, if importing Civis file(s).
> >
> >   - **storage_path** [dict::]
> >
> >   * **storage_host_id** [integer] The ID of the source storage host.
> >
> >   * **credential_id** [integer] The ID of the credentials for the source storage host.
> >
> >   * **file_paths** [list] The file or directory path(s) within the bucket from which to import. E.g. the file_path for "s3://mybucket/files/all/" would be "/files/all/"If specifying a directory path, the job will import every file found under that path. All files must have the same column layout and file format (e.g., compression, columnDelimiter, etc.).
> >
> > - **destination** [dict::]
> >
> >   - **schema** [string] The destination schema name.
> >
> >   - **table** [string] The destination table name.
> >
> >   - **remote_host_id** [integer] The ID of the destination database host.
> >
> >   - **credential_id** [integer] The ID of the credentials for the destination database.
> >
> >   - **primary_keys** [list] A list of column(s) which together uniquely identify a row in the destination table.These columns must not contain NULL values. If the import mode is "upsert", this field is required;see the Civis Helpdesk article on "Advanced CSV Imports via the Civis API" for more information.
> >
> >   - **last_modified_keys** [list] A list of the columns indicating a record has been updated.If the destination table does not exist, and the import mode is "upsert", this field is required.
> >
> > - **first_row_is_header** [boolean] A boolean value indicating whether or not the first row of the source file is a header row.
> > - **column_delimiter** [string] The column delimiter for the file. Valid arguments are "comma", "tab", and "pipe". Defaults to "comma".
> > - **escaped** [boolean] A boolean value indicating whether or not the source file has quotes escaped with a backslash.Defaults to false.

5.5. API Client **219**

- **compression** [string] The type of compression of the source file. Valid arguments are "gzip" and "none". Defaults to "none".
- **existing_table_rows** [string] The behavior if a destination table with the requested name already exists. One of "fail", "truncate", "append", "drop", or "upsert".Defaults to "fail".
- **max_errors** [integer] The maximum number of rows with errors to ignore before failing. This option is not supported for Postgres databases.
- **table_columns** [list::] An array of hashes corresponding to the columns in the order they appear in the source file. Each hash should have keys for database column "name" and "sqlType".This parameter is required if the table does not exist, the table is being dropped, or the columns in the source file do not appear in the same order as in the destination table.The "sqlType" key is not required when appending to an existing table. - name : string

  The column name.

    – **sql_type** [string] The SQL type of the column.

- **loosen_types** [boolean] If true, SQL types with precisions/lengths will have these values increased to accommodate data growth in future loads. Type loosening only occurs on table creation. Defaults to false.
- **execution** [string] In upsert mode, controls the movement of data in upsert mode. If set to "delayed", the data will be moved after a brief delay. If set to "immediate", the data will be moved immediately. In non-upsert modes, controls the speed at which detailed column stats appear in the data catalogue. Defaults to "delayed", to accommodate concurrent upserts to the same table and speedier non-upsert imports.
- **redshift_destination_options** [dict::]

    – **diststyle** [string] The diststyle to use for the table. One of "even", "all", or "key".

    – **distkey** [string] Distkey for this table in Redshift

    – **sortkeys** [list] Sortkeys for this table in Redshift. Please provide a maximum of two.
- **hidden** [boolean] The hidden status of the item.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

get_files_csv_runs(*id*, *run_id*)

    Check status of a run

        Parameters

            id [integer] The ID of the csv_import.

            run_id [integer] The ID of the run.

        Returns

        civis.response.Response

- **id** [integer] The ID of the run.
- **csv_import_id** [integer] The ID of the csv_import.
- **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **is_cancel_requested** [boolean] True if run cancel requested, else false.
- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.

- **error** [string] The error, if any, returned by the run.

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**
> >
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID of the run.
> > > - **import_id** [integer] The ID of the import.
> > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > - **started_at** [string/time] The time the last run started at.
> > > - **finished_at** [string/time] The time the last run completed.
> > > - **error** [string] The error, if any, returned by the run.

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

> 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'.
> > >
> > > **destination** [string, optional] If specified, returns imports with one of these destinations. It accepts a comma-separated list of remote host ids.
> > >
> > > **source** [string, optional] If specified, returns imports with one of these sources. It accepts a comma-separated list of remote host ids. 'DbSync' must be specified for 'type'.
> > >
> > > **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'.
> > >
> > > **author** [string, optional] If specified, return items from any of these authors. It accepts a comma- separated list of user IDs.
> > >
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> > >
> > > **archived** [string, optional] The archival status of the requested item(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**
> >
> > *civis.response.PaginatedResponse*
> >
> > > - **name** [string] The name of the import.
> > > - **sync_type** [string] The type of sync to perform; one of Dbsync, AutoImport, GdocImport, GdocExport, and Salesforce.

- **source** [dict::]

    - remote_host_id : integer

    - credential_id : integer

    - **additional_credentials** [list] Array that holds additional credentials used for specific imports. For salesforce imports, the first and only element is the client credential id. For DB Syncs, the first element is an SSL private key credential id, and the second element is the corresponding public key credential id.

    - name : string

- **destination** [dict::]

    - remote_host_id : integer

    - credential_id : integer

    - **additional_credentials** [list] Array that holds additional credentials used for specific imports. For salesforce imports, the first and only element is the client credential id. For DB Syncs, the first element is an SSL private key credential id, and the second element is the corresponding public key credential id.

    - name : string

- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **id** [integer] The ID for the import.
- is_outbound : boolean
- **job_type** [string] The job type of this import.
- state : string
- created_at : string/date-time
- updated_at : string/date-time
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

> > > > > **– error** [string] The error message for this run, if present.
> > > • **user** [dict::]

> > > > > **– id** [integer] The ID of this user.

> > > > > **– name** [string] This user's name.

> > > > > **– username** [string] This user's username.

> > > > > **– initials** [string] This user's initials.

> > > > > **– online** [boolean] Whether this user is online.
> > > • **time_zone** [string] The time zone of this import.
> > > • **archived** [string] The archival status of the requested item(s).

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

> List batch imports
> > **Parameters**
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> > > **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 50.
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > > **order** [string, optional] The field on which to order the result set. Defaults to updated_at. Must be one of: updated_at, created_at.
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> > **Returns**
> > *civis.response.PaginatedResponse*
> > > • **id** [integer] The ID for the import.
> > > • **schema** [string] The destination schema name. This schema must already exist in Redshift.
> > > • **table** [string] The destination table name, without the schema prefix. This table must already exist in Redshift.
> > > • **remote_host_id** [integer] The ID of the destination database host.
> > > • **state** [string] The state of the run; one of "queued", "running", "succeeded", "failed", or "cancelled".
> > > • **started_at** [string/time] The time the last run started at.
> > > • **finished_at** [string/time] The time the last run completed.
> > > • **error** [string] The error returned by the run, if any.

list_dependencies(*id*, *, *user_id='DEFAULT'*)

> List dependent objects for this object
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_id** [integer, optional] ID of target user
> > **Returns**
> > *civis.response.Response*
> > > • **object_type** [string] Dependent object type
> > > • **fco_type** [string] Human readable dependent object type
> > > • **id** [integer] Dependent object ID
> > > • **name** [string] Dependent object name, or nil if the requesting user cannot

read this object
- **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
- **shareable** [boolean] Whether or not the requesting user can share this object.

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

List runs for the given csv_import

Parameters

id [integer] The ID of the csv_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

civis.response.PaginatedResponse

- **id** [integer] The ID of the run.
- **csv_import_id** [integer] The ID of the csv_import.
- **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **is_cancel_requested** [boolean] True if run cancel requested, else false.
- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.
- **error** [string] The error, if any, returned by the run.

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

Get the logs for a run

Parameters

id [integer] The ID of the csv_import.

run_id [integer] The ID of the run.

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

limit [integer, optional] The maximum number of log messages to return. Default of 10000.

Returns

civis.response.Response

- **id** [integer] The ID of the log.
- **created_at** [string/date-time] The time the log was created.
- **message** [string] The log message.
- **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

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

List runs for the given import

Parameters

> **id** [integer] The ID of the import.
>
> **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 100.
>
> **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
>
> **order** [string, optional] The field on which to order the result set. Defaults to id. Must be one of: id.
>
> **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
>
> **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
>
> **Returns**
>
> *civis.response.PaginatedResponse*
>
> > • **id** [integer] The ID of the run.
> > • **import_id** [integer] The ID of the import.
> > • **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > • **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > • **started_at** [string/time] The time the last run started at.
> > • **finished_at** [string/time] The time the last run completed.
> > • **error** [string] The error, if any, returned by the run.

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

> Get the logs for a run
>
> **Parameters**
>
> > **id** [integer] The ID of the import.
> >
> > **run_id** [integer] The ID of the run.
> >
> > **last_id** [integer, optional] The ID of the last log message received. Log entries with this ID value or lower will be omitted.Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.
> >
> > **limit** [integer, optional] The maximum number of log messages to return. Default of 10000.
>
> **Returns**
>
> *civis.response.Response*
>
> > • **id** [integer] The ID of the log.
> > • **created_at** [string/date-time] The time the log was created.
> > • **message** [string] The log message.
> > • **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

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

> List the projects an Import belongs to
>
> **Parameters**
>
> > **id** [integer] The ID of the Import.
> >
> > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
>
> **Returns**
>
> *civis.response.Response*
>
> > • **id** [integer] The ID for this project.
> > • **author** [dict::]
> >
> > > – **id** [integer] The ID of this user.
> > >
> > > – **name** [string] This user's name.

- – **username** [string] This user's username.

- – **initials** [string] This user's initials.

- – **online** [boolean] Whether this user is online.
- **name** [string] The name of this project.
- **description** [string] A description of the project.
- **users** [list::] Users who can see the project. - id : integer

  The ID of this user.

  - – **name** [string] This user's name.

  - – **username** [string] This user's username.

  - – **initials** [string] This user's initials.

  - – **online** [boolean] Whether this user is online.

- auto_share : boolean
- created_at : string/time
- updated_at : string/time
- **archived** [string] The archival status of the requested item(s).

list_runs(*id*)

> Get the run history of this import
>
> > **Parameters**
> >
> > > **id** [integer]
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > - id : integer
> > > - state : string
> > > - **created_at** [string/time] The time that the run was queued.
> > > - **started_at** [string/time] The time that the run started.
> > > - **finished_at** [string/time] The time that the run completed.
> > > - **error** [string] The error message for this run, if present.

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

> Get the logs for a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the import.
> > >
> > > **run_id** [integer] The ID of the run.
> > >
> > > **last_id** [integer, optional] The ID of the last log message received. Log entries with this ID value or lower will be omitted.Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.
> > >
> > > **limit** [integer, optional] The maximum number of log messages to return. Default of 10000.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID of the log.
> > > - **created_at** [string/date-time] The time the log was created.
> > > - **message** [string] The log message.
> > > - **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

list_shares(*id*)

> List users and groups permissioned on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.

> **Returns**
> [*civis.response.Response*](#)
>> • **readers** [dict::]
>>> – **users** [list::]
>>>> * id : integer
>>>>
>>>> * name : string
>>>
>>> – **groups** [list::]
>>>> * id : integer
>>>>
>>>> * name : string
>>
>> • **writers** [dict::]
>>> – **users** [list::]
>>>> * id : integer
>>>>
>>>> * name : string
>>>
>>> – **groups** [list::]
>>>> * id : integer
>>>>
>>>> * name : string
>>
>> • **owners** [dict::]
>>> – **users** [list::]
>>>> * id : integer
>>>>
>>>> * name : string
>>>
>>> – **groups** [list::]
>>>> * id : integer
>>>>
>>>> * name : string
>>
>> • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
>> • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch_files_csv(*id*, ***, *name='DEFAULT'*, *source='DEFAULT'*, *destination='DEFAULT'*, *first_row_is_header='DEFAULT'*, *column_delimiter='DEFAULT'*, *escaped='DEFAULT'*, *compression='DEFAULT'*, *existing_table_rows='DEFAULT'*, *max_errors='DEFAULT'*, *table_columns='DEFAULT'*, *loosen_types='DEFAULT'*, *execution='DEFAULT'*, *redshift_destination_options='DEFAULT'*)

> Update some attributes of this CSV Import
>> **Parameters**
>>> **id** [integer] The ID for the import.
>>> **name** [string, optional] The name of the import.
>>> **source** [dict, optional::]
>>>> • **file_ids** [list] The file ID(s) to import, if importing Civis file(s).
>>>> • **storage_path** [dict::]
>>>>> – **storage_host_id** [integer] The ID of the source storage host.
>>>>>
>>>>> – **credential_id** [integer] The ID of the credentials for the source storage host.

- **file_paths** [list] The file or directory path(s) within the bucket from which to import. E.g. the file_path for "s3://mybucket/files/all/" would be "/files/all/"If specifying a directory path, the job will import every file found under that path. All files must have the same column layout and file format (e.g., compression, columnDelimiter, etc.).

destination [dict, optional::]
- **schema** [string] The destination schema name.
- **table** [string] The destination table name.
- **remote_host_id** [integer] The ID of the destination database host.
- **credential_id** [integer] The ID of the credentials for the destination database.
- **primary_keys** [list] A list of column(s) which together uniquely identify a row in the destination table.These columns must not contain NULL values. If the import mode is "upsert", this field is required;see the Civis Helpdesk article on "Advanced CSV Imports via the Civis API" for more information.
- **last_modified_keys** [list] A list of the columns indicating a record has been updated.If the destination table does not exist, and the import mode is "upsert", this field is required.

first_row_is_header [boolean, optional] A boolean value indicating whether or not the first row of the source file is a header row.

column_delimiter [string, optional] The column delimiter for the file. Valid arguments are "comma", "tab", and "pipe". Defaults to "comma".

escaped [boolean, optional] A boolean value indicating whether or not the source file has quotes escaped with a backslash.Defaults to false.

compression [string, optional] The type of compression of the source file. Valid arguments are "gzip" and "none". Defaults to "none".

existing_table_rows [string, optional] The behavior if a destination table with the requested name already exists. One of "fail", "truncate", "append", "drop", or "upsert".Defaults to "fail".

max_errors [integer, optional] The maximum number of rows with errors to ignore before failing. This option is not supported for Postgres databases.

table_columns [list, optional::] An array of hashes corresponding to the columns in the order they appear in the source file. Each hash should have keys for database column "name" and "sqlType".This parameter is required if the table does not exist, the table is being dropped, or the columns in the source file do not appear in the same order as in the destination table.The "sqlType" key is not required when appending to an existing table. - name : string
    The column name.

- **sql_type** [string] The SQL type of the column.

loosen_types [boolean, optional] If true, SQL types with precisions/lengths will have these values increased to accommodate data growth in future loads. Type loosening only occurs on table creation. Defaults to false.

execution [string, optional] In upsert mode, controls the movement of data in upsert mode. If set to "delayed", the data will be moved after a brief delay. If set to "immediate", the data will be moved immediately. In non-upsert modes, controls the speed at which detailed column stats appear in the data catalogue. Defaults to "delayed", to accommodate concurrent upserts to the same table and speedier non-upsert imports.

redshift_destination_options [dict, optional::]
- **diststyle** [string] The diststyle to use for the table. One of "even", "all",

> or "key".

- **distkey** [string] Distkey for this table in Redshift
- **sortkeys** [list] Sortkeys for this table in Redshift. Please provide a maximum of two.

Returns

    civis.response.Response

- **id** [integer] The ID for the import.
- **name** [string] The name of the import.
- **source** [dict::]

    – **file_ids** [list] The file ID(s) to import, if importing Civis file(s).

    – **storage_path** [dict::]

        ∗ **storage_host_id** [integer] The ID of the source storage host.

        ∗ **credential_id** [integer] The ID of the credentials for the source storage host.

        ∗ **file_paths** [list] The file or directory path(s) within the bucket from which to import. E.g. the file_path for "s3://mybucket/files/all/" would be "/files/all/"If specifying a directory path, the job will import every file found under that path. All files must have the same column layout and file format (e.g., compression, columnDelimiter, etc.).

- **destination** [dict::]

    – **schema** [string] The destination schema name.

    – **table** [string] The destination table name.

    – **remote_host_id** [integer] The ID of the destination database host.

    – **credential_id** [integer] The ID of the credentials for the destination database.

    – **primary_keys** [list] A list of column(s) which together uniquely identify a row in the destination table.These columns must not contain NULL values. If the import mode is "upsert", this field is required;see the Civis Helpdesk article on "Advanced CSV Imports via the Civis API" for more information.

    – **last_modified_keys** [list] A list of the columns indicating a record has been updated.If the destination table does not exist, and the import mode is "upsert", this field is required.

- **first_row_is_header** [boolean] A boolean value indicating whether or not the first row of the source file is a header row.
- **column_delimiter** [string] The column delimiter for the file. Valid arguments are "comma", "tab", and "pipe". Defaults to "comma".
- **escaped** [boolean] A boolean value indicating whether or not the source file has quotes escaped with a backslash.Defaults to false.

- **compression** [string] The type of compression of the source file. Valid
  arguments are "gzip" and "none". Defaults to "none".
- **existing_table_rows** [string] The behavior if a destination table with the
  requested name already exists. One of "fail", "truncate", "append",
  "drop", or "upsert".Defaults to "fail".
- **max_errors** [integer] The maximum number of rows with errors to ignore
  before failing. This option is not supported for Postgres databases.
- **table_columns** [list::] An array of hashes corresponding to the columns
  in the order they appear in the source file. Each hash should have
  keys for database column "name" and "sqlType".This parameter is
  required if the table does not exist, the table is being dropped, or the
  columns in the source file do not appear in the same order as in the
  destination table.The "sqlType" key is not required when appending
  to an existing table. - name : string

    The column name.

    – **sql_type** [string] The SQL type of the column.
- **loosen_types** [boolean] If true, SQL types with precisions/lengths will
  have these values increased to accommodate data growth in future
  loads. Type loosening only occurs on table creation. Defaults to
  false.
- **execution** [string] In upsert mode, controls the movement of data in up-
  sert mode. If set to "delayed", the data will be moved after a brief
  delay. If set to "immediate", the data will be moved immediately.
  In non-upsert modes, controls the speed at which detailed column
  stats appear in the data catalogue. Defaults to "delayed", to accom-
  modate concurrent upserts to the same table and speedier non-upsert
  imports.
- **redshift_destination_options** [dict::]

    – **diststyle** [string] The diststyle to use for the table. One of
      "even", "all", or "key".

    – **distkey** [string] Distkey for this table in Redshift

    – **sortkeys** [list] Sortkeys for this table in Redshift. Please pro-
      vide a maximum of two.
- **hidden** [boolean] The hidden status of the item.
- **my_permission_level** [string] Your permission level on the object. One
  of "read", "write", or "manage".

post(*name*, *sync_type*, *is_outbound*, *, *source='DEFAULT'*, *destination='DEFAULT'*, *schedule='DEFAULT'*,
  notifications='DEFAULT', *parent_id='DEFAULT'*, *next_run_at='DEFAULT'*, *time_zone='DEFAULT'*,
  hidden='DEFAULT')
  Create a new import configuration
    Parameters
      name [string] The name of the import.
      sync_type [string] The type of sync to perform; one of Dbsync, AutoImport, GdocIm-
        port, GdocExport, and Salesforce.
      is_outbound [boolean]
      source [dict, optional::]
        - remote_host_id : integer
        - credential_id : integer
        - **additional_credentials** [list] Array that holds additional credentials used
          for specific imports. For salesforce imports, the first and only el-

> element is the client credential id. For DB Syncs, the first element
> is an SSL private key credential id, and the second element is the
> corresponding public key credential id.

destination [dict, optional::]
- remote_host_id : integer
- credential_id : integer
- **additional_credentials** [list] Array that holds additional credentials used
  for specific imports. For salesforce imports, the first and only el-
  ement is the client credential id. For DB Syncs, the first element
  is an SSL private key credential id, and the second element is the
  corresponding public key credential id.

schedule [dict, optional::]
- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting
  at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes,
  number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on,
  mutually exclusive with scheduledDays.

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

parent_id [integer, optional] Parent id to trigger this import from
next_run_at [string/time, optional] The time of the next scheduled run.
time_zone [string, optional] The time zone of this import.
hidden [boolean, optional] The hidden status of the item.

Returns

civis.response.Response
- **name** [string] The name of the import.
- **sync_type** [string] The type of sync to perform; one of Dbsync, AutoIm-
  port, GdocImport, GdocExport, and Salesforce.
- **source** [dict::]

    - remote_host_id : integer

    - credential_id : integer

    - **additional_credentials** [list] Array that holds additional cre-
      dentials used for specific imports. For salesforce im-

ports, the first and only element is the client credential id. For DB Syncs, the first element is an SSL private key credential id, and the second element is the corresponding public key credential id.

- name : string

- **destination** [dict::]

    - remote_host_id : integer

    - credential_id : integer

    - **additional_credentials** [list] Array that holds additional credentials used for specific imports. For salesforce imports, the first and only element is the client credential id. For DB Syncs, the first element is an SSL private key credential id, and the second element is the corresponding public key credential id.

    - name : string

- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

> > > > – **failure_on** [boolean] If failure email notifications are on.
> > > - **parent_id** [integer] Parent id to trigger this import from
> > > - **id** [integer] The ID for the import.
> > > - is_outbound : boolean
> > > - **job_type** [string] The job type of this import.
> > > - **syncs** [list::] List of syncs. - id : integer - source : dict:

```
- id : integer
    The ID of the table or file, if available.
- path : string
    The path of the dataset to sync from; for a␣
→database source,
    schema.tablename. If you are doing a Google␣
→Sheet export, this
    can be blank. This is a legacy parameter, it␣
→is recommended you
    use one of the following: databaseTable, file,␣
→googleWorksheet,
    salesforce
- database_table : dict::
    - schema : string
        The database schema name.
    - table : string
        The database table name.
    - use_without_schema : boolean
        This attribute is no longer available;␣
→defaults to false
        but cannot be used.
- file : dict::
    - id : integer
        The file id.
- google_worksheet : dict::
    - spreadsheet : string
        The spreadsheet document name.
    - spreadsheet_id : string
        The spreadsheet document id.
    - worksheet : string
        The worksheet tab name.
    - worksheet_id : integer
        The worksheet tab id.
- salesforce : dict::
    - object_name : string
        The Salesforce object name.
```

> > > > – **destination** [dict::]

> > > > > ∗ **path** [string] The schema.tablename to sync to. If you are doing a Google Sheet export, this is the spreadsheet and sheet name separated by a period. i.e. if you have a spreadsheet named "MySpreadsheet" and a sheet called "Sheet1" this field would be "MySpreadsheet.Sheet1". This is a legacy parameter, it is recommended you use one of the following: databaseTable,

googleWorksheet

* **database_table** [dict::]

    · **schema** [string] The database schema name.

    · **table** [string] The database table name.

    · **use_without_schema** [boolean] This at-
        tribute is no longer available; defaults to
        false but cannot be used.

* **google_worksheet** [dict::]

    · **spreadsheet** [string] The spreadsheet docu-
        ment name.

    · **spreadsheet_id** [string] The spreadsheet
        document id.

    · **worksheet** [string] The worksheet tab
        name.

    · **worksheet_id** [integer] The worksheet tab
        id.

– **advanced_options** [dict::]

    * max_errors : integer

    * existing_table_rows : string

    * diststyle : string

    * distkey : string

    * sortkey1 : string

    * sortkey2 : string

    * column_delimiter : string

    * **column_overrides** [dict] Hash used for overriding
        auto-detected names and types, with keys being
        the index of the column being overridden.

    * **escaped** [boolean] If true, escape quotes with a
        backslash; otherwise, escape quotes by double-
        quoting. Defaults to false.

    * identity_column : string

    * row_chunk_size : integer

    * wipe_destination_table : boolean

    * truncate_long_lines : boolean

    * invalid_char_replacement : string

    * verify_table_row_counts : boolean

    * **partition_column_name** [string] This parameter
        is deprecated

    * **partition_schema_name** [string] This parameter
        is deprecated

* **partition_table_name** [string] This parameter is deprecated

* **partition_table_partition_column_min_name** [string] This parameter is deprecated

* **partition_table_partition_column_max_name** [string] This parameter is deprecated

* last_modified_column : string

* **mysql_catalog_matches_schema** [boolean] This attribute is no longer available; defaults to true but cannot be used.

* **chunking_method** [string] This parameter is deprecated

* first_row_is_header : boolean

* **export_action** [string] The kind of export action you want to have the export execute. Set to "newsprsht" if you want a new worksheet inside a new spreadsheet. Set to "newwksht" if you want a new worksheet inside an existing spreadsheet. Set to "updatewksht" if you want to overwrite an existing worksheet inside an existing spreadsheet. Set to "appendwksht" if you want to append to the end of an existing worksheet inside an existing spreadsheet. Default is set to "newsprsht"

* **sql_query** [string] If you are doing a Google Sheet export, this is your SQL query.

* contact_lists : string

* soql_query : string

* include_deleted_records : boolean

- state : string
- created_at : string/date-time
- updated_at : string/date-time
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.

- **user** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this import.
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

post_batches(*file_ids*, *schema*, *table*, *remote_host_id*, *credential_id*, ***, *column_delimiter='DEFAULT'*, *first_row_is_header='DEFAULT'*, *compression='DEFAULT'*, *hidden='DEFAULT'*)

> Upload multiple files to Civis
>
> > **Parameters**
> >
> > > **file_ids** [list] The file IDs for the import.
> > >
> > > **schema** [string] The destination schema name. This schema must already exist in Redshift.
> > >
> > > **table** [string] The destination table name, without the schema prefix. This table must already exist in Redshift.
> > >
> > > **remote_host_id** [integer] The ID of the destination database host.
> > >
> > > **credential_id** [integer] The ID of the credentials to be used when performing the database import.
> > >
> > > **column_delimiter** [string, optional] The column delimiter for the file. Valid arguments are "comma", "tab", and "pipe". If unspecified, defaults to "comma".
> > >
> > > **first_row_is_header** [boolean, optional] A boolean value indicating whether or not the first row is a header row. If unspecified, defaults to false.
> > >
> > > **compression** [string, optional] The type of compression. Valid arguments are "gzip", "zip", and "none". If unspecified, defaults to "gzip".
> > >
> > > **hidden** [boolean, optional] The hidden status of the item.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **id** [integer] The ID for the import.
> > > > - **schema** [string] The destination schema name. This schema must already exist in Redshift.
> > > > - **table** [string] The destination table name, without the schema prefix. This table must already exist in Redshift.
> > > > - **remote_host_id** [integer] The ID of the destination database host.
> > > > - **state** [string] The state of the run; one of "queued", "running", "succeeded", "failed", or "cancelled".
> > > > - **started_at** [string/time] The time the last run started at.
> > > > - **finished_at** [string/time] The time the last run completed.
> > > > - **error** [string] The error returned by the run, if any.
> > > > - **hidden** [boolean] The hidden status of the item.

post_cancel(*id*)

> Cancel a run
>
> > **Parameters**

> > > > **id** [integer] The ID of the job.
> > >
> > > **Returns**
> > >
> > > > *civis.response.Response*
> > > >
> > > > > - **id** [integer] The ID of the run.
> > > > > - **state** [string] The state of the run, one of 'queued', 'running' or 'cancelled'.
> > > > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.

> **post_files**(*schema*, *name*, *remote_host_id*, *credential_id*, *, *max_errors='DEFAULT'*,
> > *existing_table_rows='DEFAULT'*, *diststyle='DEFAULT'*, *distkey='DEFAULT'*,
> > *sortkey1='DEFAULT'*, *sortkey2='DEFAULT'*, *column_delimiter='DEFAULT'*,
> > *first_row_is_header='DEFAULT'*, *multipart='DEFAULT'*, *escaped='DEFAULT'*,
> > *hidden='DEFAULT'*)
>
> > Initate an import of a tabular file into the platform
> >
> > > **Parameters**
> > >
> > > > **schema** [string] The schema of the destination table.
> > > >
> > > > **name** [string] The name of the destination table.
> > > >
> > > > **remote_host_id** [integer] The id of the destination database host.
> > > >
> > > > **credential_id** [integer] The id of the credentials to be used when performing the database import.
> > > >
> > > > **max_errors** [integer, optional] The maximum number of rows with errors to remove from the import before failing.
> > > >
> > > > **existing_table_rows** [string, optional] The behaviour if a table with the requested name already exists. One of "fail", "truncate", "append", or "drop".Defaults to "fail".
> > > >
> > > > **diststyle** [string, optional] The diststyle to use for the table. One of "even", "all", or "key".
> > > >
> > > > **distkey** [string, optional] The column to use as the distkey for the table.
> > > >
> > > > **sortkey1** [string, optional] The column to use as the sort key for the table.
> > > >
> > > > **sortkey2** [string, optional] The second column in a compound sortkey for the table.
> > > >
> > > > **column_delimiter** [string, optional] The column delimiter of the file. If column_delimiter is null or omitted, it will be auto-detected. Valid arguments are "comma", "tab", and "pipe".
> > > >
> > > > **first_row_is_header** [boolean, optional] A boolean value indicating whether or not the first row is a header row. If first_row_is_header is null or omitted, it will be auto-detected.
> > > >
> > > > **multipart** [boolean, optional] If true, the upload URI will require a *multipart/form-data* POST request. Defaults to false.
> > > >
> > > > **escaped** [boolean, optional] If true, escape quotes with a backslash; otherwise, escape quotes by double-quoting. Defaults to false.
> > > >
> > > > **hidden** [boolean, optional] The hidden status of the item.
> > >
> > > **Returns**
> > >
> > > > *civis.response.Response*
> > > >
> > > > > - **id** [integer] The id of the import.
> > > > > - **upload_uri** [string] The URI which may be used to upload a tabular file for import. You must use this URI to upload the file you wish imported and then inform the Civis API when your upload is complete using the URI given by the runUri field of this response.
> > > > > - **run_uri** [string] The URI to POST to once the file upload is complete. After uploading the file using the URI given in the uploadUri attribute of the response, POST to this URI to initiate the import of your uploaded file into the platform.
> > > > > - **upload_fields** [dict] If multipart was set to true, these fields should be included in the multipart upload.

---

post_files_csv(*source*, *destination*, *first_row_is_header*, *, *name='DEFAULT'*,
               column_delimiter='DEFAULT', *escaped='DEFAULT'*, *compression='DEFAULT'*,
               existing_table_rows='DEFAULT', *max_errors='DEFAULT'*, *table_columns='DEFAULT'*,
               loosen_types='DEFAULT', *execution='DEFAULT'*,
               redshift_destination_options='DEFAULT', *hidden='DEFAULT'*)

    Create a CSV Import

        Parameters

            source [dict::]

                • **file_ids** [list] The file ID(s) to import, if importing Civis file(s).

                • **storage_path** [dict::]

                            – **storage_host_id** [integer] The ID of the source storage host.

                            – **credential_id** [integer] The ID of the credentials for the source storage host.

                            – **file_paths** [list] The file or directory path(s) within the bucket from which to import. E.g. the file_path for "s3://mybucket/files/all/" would be "/files/all/"If specifying a directory path, the job will import every file found under that path. All files must have the same column layout and file format (e.g., compression, columnDelimiter, etc.).

            destination [dict::]

                • **schema** [string] The destination schema name.

                • **table** [string] The destination table name.

                • **remote_host_id** [integer] The ID of the destination database host.

                • **credential_id** [integer] The ID of the credentials for the destination database.

                • **primary_keys** [list] A list of column(s) which together uniquely identify a row in the destination table.These columns must not contain NULL values. If the import mode is "upsert", this field is required;see the Civis Helpdesk article on "Advanced CSV Imports via the Civis API" for more information.

                • **last_modified_keys** [list] A list of the columns indicating a record has been updated.If the destination table does not exist, and the import mode is "upsert", this field is required.

            first_row_is_header [boolean] A boolean value indicating whether or not the first row of the source file is a header row.

            name [string, optional] The name of the import.

            column_delimiter [string, optional] The column delimiter for the file. Valid arguments are "comma", "tab", and "pipe". Defaults to "comma".

            escaped [boolean, optional] A boolean value indicating whether or not the source file has quotes escaped with a backslash.Defaults to false.

            compression [string, optional] The type of compression of the source file. Valid arguments are "gzip" and "none". Defaults to "none".

            existing_table_rows [string, optional] The behavior if a destination table with the requested name already exists. One of "fail", "truncate", "append", "drop", or "upsert".Defaults to "fail".

            max_errors [integer, optional] The maximum number of rows with errors to ignore before failing. This option is not supported for Postgres databases.

            table_columns [list, optional::] An array of hashes corresponding to the columns in the order they appear in the source file. Each hash should have keys for database column "name" and "sqlType".This parameter is required if the table does not exist, the table is being dropped, or the columns in the source file do not appear

---

in the same order as in the destination table.The "sqlType" key is not required when appending to an existing table. - name : string

The column name.

- **sql_type** [string] The SQL type of the column.

loosen_types [boolean, optional] If true, SQL types with precisions/lengths will have these values increased to accommodate data growth in future loads. Type loosening only occurs on table creation. Defaults to false.

execution [string, optional] In upsert mode, controls the movement of data in upsert mode. If set to "delayed", the data will be moved after a brief delay. If set to "immediate", the data will be moved immediately. In non-upsert modes, controls the speed at which detailed column stats appear in the data catalogue. Defaults to "delayed", to accommodate concurrent upserts to the same table and speedier non-upsert imports.

redshift_destination_options [dict, optional::]

- **diststyle** [string] The diststyle to use for the table. One of "even", "all", or "key".
- **distkey** [string] Distkey for this table in Redshift
- **sortkeys** [list] Sortkeys for this table in Redshift. Please provide a maximum of two.

hidden [boolean, optional] The hidden status of the item.

Returns

civis.response.Response

- **id** [integer] The ID for the import.
- **name** [string] The name of the import.
- **source** [dict::]

  – **file_ids** [list] The file ID(s) to import, if importing Civis file(s).

  – **storage_path** [dict::]

    * **storage_host_id** [integer] The ID of the source storage host.

    * **credential_id** [integer] The ID of the credentials for the source storage host.

    * **file_paths** [list] The file or directory path(s) within the bucket from which to import. E.g. the file_path for "s3://mybucket/files/all/" would be "/files/all/"If specifying a directory path, the job will import every file found under that path. All files must have the same column layout and file format (e.g., compression, columnDelimiter, etc.).

- **destination** [dict::]

  – **schema** [string] The destination schema name.

  – **table** [string] The destination table name.

  – **remote_host_id** [integer] The ID of the destination database host.

  – **credential_id** [integer] The ID of the credentials for the destination database.

  – **primary_keys** [list] A list of column(s) which together

> > > uniquely identify a row in the destination table.These columns must not contain NULL values. If the import mode is "upsert", this field is required;see the Civis Helpdesk article on "Advanced CSV Imports via the Civis API" for more information.

- **last_modified_keys** [list] A list of the columns indicating a record has been updated.If the destination table does not exist, and the import mode is "upsert", this field is required.

- **first_row_is_header** [boolean] A boolean value indicating whether or not the first row of the source file is a header row.
- **column_delimiter** [string] The column delimiter for the file. Valid arguments are "comma", "tab", and "pipe". Defaults to "comma".
- **escaped** [boolean] A boolean value indicating whether or not the source file has quotes escaped with a backslash.Defaults to false.
- **compression** [string] The type of compression of the source file. Valid arguments are "gzip" and "none". Defaults to "none".
- **existing_table_rows** [string] The behavior if a destination table with the requested name already exists. One of "fail", "truncate", "append", "drop", or "upsert".Defaults to "fail".
- **max_errors** [integer] The maximum number of rows with errors to ignore before failing. This option is not supported for Postgres databases.
- **table_columns** [list::] An array of hashes corresponding to the columns in the order they appear in the source file. Each hash should have keys for database column "name" and "sqlType".This parameter is required if the table does not exist, the table is being dropped, or the columns in the source file do not appear in the same order as in the destination table.The "sqlType" key is not required when appending to an existing table. - name : string

  The column name.

    - **sql_type** [string] The SQL type of the column.

- **loosen_types** [boolean] If true, SQL types with precisions/lengths will have these values increased to accommodate data growth in future loads. Type loosening only occurs on table creation. Defaults to false.
- **execution** [string] In upsert mode, controls the movement of data in upsert mode. If set to "delayed", the data will be moved after a brief delay. If set to "immediate", the data will be moved immediately. In non-upsert modes, controls the speed at which detailed column stats appear in the data catalogue. Defaults to "delayed", to accommodate concurrent upserts to the same table and speedier non-upsert imports.
- **redshift_destination_options** [dict::]

    - **diststyle** [string] The diststyle to use for the table. One of "even", "all", or "key".

    - **distkey** [string] Distkey for this table in Redshift

    - **sortkeys** [list] Sortkeys for this table in Redshift. Please provide a maximum of two.

- **hidden** [boolean] The hidden status of the item.

> - **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

post_files_csv_runs(*id*)

> Start a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the csv_import.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID of the run.
> > > - **csv_import_id** [integer] The ID of the csv_import.
> > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > - **started_at** [string/time] The time the last run started at.
> > > - **finished_at** [string/time] The time the last run completed.
> > > - **error** [string] The error, if any, returned by the run.

post_files_runs(*id*)

> Start a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the import.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID of the run.
> > > - **import_id** [integer] The ID of the import.
> > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > - **started_at** [string/time] The time the last run started at.
> > > - **finished_at** [string/time] The time the last run completed.
> > > - **error** [string] The error, if any, returned by the run.

post_runs(*id*)

> Run an import
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the import to run.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > - **run_id** [integer] The ID of the new run triggered.

post_syncs(*id*, *source*, *destination*, ***, *advanced_options='DEFAULT'*)

> Create a sync
>
> > **Parameters**
> >
> > > **id** [integer]
> > >
> > > **source** [dict::]
> > >
> > > > - **path** [string] The path of the dataset to sync from; for a database source, schema.tablename. If you are doing a Google Sheet export, this can be blank. This is a legacy parameter, it is recommended you use one of the following: databaseTable, file, googleWorksheet, salesforce
> > > > - **database_table** [dict::]
> > > >
> > > > > – **schema** [string] The database schema name.
> > > > >
> > > > > – **table** [string] The database table name.
> > > > >
> > > > > – **use_without_schema** [boolean] This attribute is no longer

> available; defaults to false but cannot be used.
- file : dict
- **google_worksheet** [dict::]

    – **spreadsheet** [string] The spreadsheet document name.

    – **spreadsheet_id** [string] The spreadsheet document id.

    – **worksheet** [string] The worksheet tab name.

    – **worksheet_id** [integer] The worksheet tab id.
- **salesforce** [dict::]

    – **object_name** [string] The Salesforce object name.

destination [dict::]
- **path** [string] The schema.tablename to sync to. If you are doing a Google Sheet export, this is the spreadsheet and sheet name separated by a period. i.e. if you have a spreadsheet named "MySpreadsheet" and a sheet called "Sheet1" this field would be "MySpreadsheet.Sheet1". This is a legacy parameter, it is recommended you use one of the following: databaseTable, googleWorksheet
- **database_table** [dict::]

    – **schema** [string] The database schema name.

    – **table** [string] The database table name.

    – **use_without_schema** [boolean] This attribute is no longer available; defaults to false but cannot be used.
- **google_worksheet** [dict::]

    – **spreadsheet** [string] The spreadsheet document name.

    – **spreadsheet_id** [string] The spreadsheet document id.

    – **worksheet** [string] The worksheet tab name.

    – **worksheet_id** [integer] The worksheet tab id.

advanced_options [dict, optional::]
- max_errors : integer
- existing_table_rows : string
- diststyle : string
- distkey : string
- sortkey1 : string
- sortkey2 : string
- column_delimiter : string
- **column_overrides** [dict] Hash used for overriding auto-detected names and types, with keys being the index of the column being overridden.
- **escaped** [boolean] If true, escape quotes with a backslash; otherwise, escape quotes by double-quoting. Defaults to false.
- identity_column : string
- row_chunk_size : integer
- wipe_destination_table : boolean
- truncate_long_lines : boolean
- invalid_char_replacement : string
- verify_table_row_counts : boolean
- **partition_column_name** [string] This parameter is deprecated
- **partition_schema_name** [string] This parameter is deprecated
- **partition_table_name** [string] This parameter is deprecated

- **partition_table_partition_column_min_name** [string] This parameter is deprecated
- **partition_table_partition_column_max_name** [string] This parameter is deprecated
- last_modified_column : string
- **mysql_catalog_matches_schema** [boolean] This attribute is no longer available; defaults to true but cannot be used.
- **chunking_method** [string] This parameter is deprecated
- first_row_is_header : boolean
- **export_action** [string] The kind of export action you want to have the export execute. Set to "newsprsht" if you want a new worksheet inside a new spreadsheet. Set to "newwksht" if you want a new worksheet inside an existing spreadsheet. Set to "updatewksht" if you want to overwrite an existing worksheet inside an existing spreadsheet. Set to "appendwksht" if you want to append to the end of an existing worksheet inside an existing spreadsheet. Default is set to "newsprsht"
- **sql_query** [string] If you are doing a Google Sheet export, this is your SQL query.
- contact_lists : string
- soql_query : string
- include_deleted_records : boolean

Returns

[*civis.response.Response*](#)

- id : integer
- **source** [dict::]

    – **id** [integer] The ID of the table or file, if available.

    – **path** [string] The path of the dataset to sync from; for a database source, schema.tablename. If you are doing a Google Sheet export, this can be blank. This is a legacy parameter, it is recommended you use one of the following: databaseTable, file, googleWorksheet, salesforce

    – **database_table** [dict::]

        * **schema** [string] The database schema name.

        * **table** [string] The database table name.

        * **use_without_schema** [boolean] This attribute is no longer available; defaults to false but cannot be used.

    – **file** [dict::]

        * **id** [integer] The file id.

    – **google_worksheet** [dict::]

        * **spreadsheet** [string] The spreadsheet document name.

        * **spreadsheet_id** [string] The spreadsheet document id.

        * **worksheet** [string] The worksheet tab name.

        * **worksheet_id** [integer] The worksheet tab id.

- **salesforce** [dict::]

    * **object_name** [string] The Salesforce object name.
- **destination** [dict::]

    - **path** [string] The schema.tablename to sync to. If you are doing a Google Sheet export, this is the spreadsheet and sheet name separated by a period. i.e. if you have a spreadsheet named "MySpreadsheet" and a sheet called "Sheet1" this field would be "MySpreadsheet.Sheet1". This is a legacy parameter, it is recommended you use one of the following: databaseTable, googleWorksheet

    - **database_table** [dict::]

        * **schema** [string] The database schema name.

        * **table** [string] The database table name.

        * **use_without_schema** [boolean] This attribute is no longer available; defaults to false but cannot be used.

    - **google_worksheet** [dict::]

        * **spreadsheet** [string] The spreadsheet document name.

        * **spreadsheet_id** [string] The spreadsheet document id.

        * **worksheet** [string] The worksheet tab name.

        * **worksheet_id** [integer] The worksheet tab id.
- **advanced_options** [dict::]

    - max_errors : integer

    - existing_table_rows : string

    - diststyle : string

    - distkey : string

    - sortkey1 : string

    - sortkey2 : string

    - column_delimiter : string

    - **column_overrides** [dict] Hash used for overriding auto-detected names and types, with keys being the index of the column being overridden.

    - **escaped** [boolean] If true, escape quotes with a backslash; otherwise, escape quotes by double-quoting. Defaults to false.

    - identity_column : string

    - row_chunk_size : integer

    - wipe_destination_table : boolean

    - truncate_long_lines : boolean

- invalid_char_replacement : string

- verify_table_row_counts : boolean

- **partition_column_name** [string] This parameter is deprecated

- **partition_schema_name** [string] This parameter is deprecated

- **partition_table_name** [string] This parameter is deprecated

- **partition_table_partition_column_min_name** [string] This parameter is deprecated

- **partition_table_partition_column_max_name** [string] This parameter is deprecated

- last_modified_column : string

- **mysql_catalog_matches_schema** [boolean] This attribute is no longer available; defaults to true but cannot be used.

- **chunking_method** [string] This parameter is deprecated

- first_row_is_header : boolean

- **export_action** [string] The kind of export action you want to have the export execute. Set to "newsprsht" if you want a new worksheet inside a new spreadsheet. Set to "newwksht" if you want a new worksheet inside an existing spreadsheet. Set to "updatewksht" if you want to overwrite an existing worksheet inside an existing spreadsheet. Set to "appendwksht" if you want to append to the end of an existing worksheet inside an existing spreadsheet. Default is set to "newsprsht"

- **sql_query** [string] If you are doing a Google Sheet export, this is your SQL query.

- contact_lists : string

- soql_query : string

- include_deleted_records : boolean

put(*id*, *name*, *sync_type*, *is_outbound*, *, *source='DEFAULT'*, *destination='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*, *parent_id='DEFAULT'*, *next_run_at='DEFAULT'*, *time_zone='DEFAULT'*)

    Update an import

        Parameters

            id [integer] The ID for the import.

            name [string] The name of the import.

            sync_type [string] The type of sync to perform; one of Dbsync, AutoImport, GdocImport, GdocExport, and Salesforce.

            is_outbound [boolean]

            source [dict, optional::]

                • remote_host_id : integer

                • credential_id : integer

                • **additional_credentials** [list] Array that holds additional credentials used for specific imports. For salesforce imports, the first and only element is the client credential id. For DB Syncs, the first element

is an SSL private key credential id, and the second element is the corresponding public key credential id.

destination [dict, optional::]

- remote_host_id : integer
- credential_id : integer
- **additional_credentials** [list] Array that holds additional credentials used for specific imports. For salesforce imports, the first and only element is the client credential id. For DB Syncs, the first element is an SSL private key credential id, and the second element is the corresponding public key credential id.

schedule [dict, optional::]

- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

notifications [dict, optional::]

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

parent_id [integer, optional] Parent id to trigger this import from

next_run_at [string/time, optional] The time of the next scheduled run.

time_zone [string, optional] The time zone of this import.

Returns

civis.response.Response

- **name** [string] The name of the import.
- **sync_type** [string] The type of sync to perform; one of Dbsync, AutoImport, GdocImport, GdocExport, and Salesforce.
- **source** [dict::]

  – remote_host_id : integer

  – credential_id : integer

  – **additional_credentials** [list] Array that holds additional credentials used for specific imports. For salesforce imports, the first and only element is the client credential id. For DB Syncs, the first element is an SSL private key

> > credential id, and the second element is the corresponding public key credential id.

> > – name : string

- **destination** [dict::]

> > – remote_host_id : integer

> > – credential_id : integer

> > – **additional_credentials** [list] Array that holds additional credentials used for specific imports. For salesforce imports, the first and only element is the client credential id. For DB Syncs, the first element is an SSL private key credential id, and the second element is the corresponding public key credential id.

> > – name : string

- **schedule** [dict::]

> > – **scheduled** [boolean] If the item is scheduled.

> > – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

> > – **scheduled_hours** [list] Hours of the day it is scheduled on.

> > – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

> > – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

> > – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

> > – **urls** [list] URLs to receive a POST request at job completion

> > – **success_email_subject** [string] Custom subject line for success e-mail.

> > – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

> > – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

> > – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

> > – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

> > – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

> > – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

> > – **success_on** [boolean] If success email notifications are on.

> > – **failure_on** [boolean] If failure email notifications are on.

- **parent_id** [integer] Parent id to trigger this import from
- **id** [integer] The ID for the import.
- is_outbound : boolean
- **job_type** [string] The job type of this import.
- **syncs** [list::] List of syncs. - id : integer - source : dict:

```
- id : integer
    The ID of the table or file, if available.
- path : string
    The path of the dataset to sync from; for a
→database source,
    schema.tablename. If you are doing a Google
→Sheet export, this
    can be blank. This is a legacy parameter, it
→is recommended you
    use one of the following: databaseTable, file,
→googleWorksheet,
    salesforce
- database_table : dict::
    - schema : string
        The database schema name.
    - table : string
        The database table name.
    - use_without_schema : boolean
        This attribute is no longer available;
→defaults to false
        but cannot be used.
- file : dict::
    - id : integer
        The file id.
- google_worksheet : dict::
    - spreadsheet : string
        The spreadsheet document name.
    - spreadsheet_id : string
        The spreadsheet document id.
    - worksheet : string
        The worksheet tab name.
    - worksheet_id : integer
        The worksheet tab id.
- salesforce : dict::
    - object_name : string
        The Salesforce object name.
```

- **destination** [dict::]

  * **path** [string] The schema.tablename to sync to. If you are doing a Google Sheet export, this is the spreadsheet and sheet name separated by a period. i.e. if you have a spreadsheet named "MySpreadsheet" and a sheet called "Sheet1" this field would be "MySpreadsheet.Sheet1". This is a legacy parameter, it is recommended you use one of the following: databaseTable, googleWorksheet

* **database_table** [dict::]

    · **schema** [string] The database schema name.

    · **table** [string] The database table name.

    · **use_without_schema** [boolean] This attribute is no longer available; defaults to false but cannot be used.

* **google_worksheet** [dict::]

    · **spreadsheet** [string] The spreadsheet document name.

    · **spreadsheet_id** [string] The spreadsheet document id.

    · **worksheet** [string] The worksheet tab name.

    · **worksheet_id** [integer] The worksheet tab id.

– **advanced_options** [dict::]

    * max_errors : integer

    * existing_table_rows : string

    * diststyle : string

    * distkey : string

    * sortkey1 : string

    * sortkey2 : string

    * column_delimiter : string

    * **column_overrides** [dict] Hash used for overriding auto-detected names and types, with keys being the index of the column being overridden.

    * **escaped** [boolean] If true, escape quotes with a backslash; otherwise, escape quotes by double-quoting. Defaults to false.

    * identity_column : string

    * row_chunk_size : integer

    * wipe_destination_table : boolean

    * truncate_long_lines : boolean

    * invalid_char_replacement : string

    * verify_table_row_counts : boolean

    * **partition_column_name** [string] This parameter is deprecated

    * **partition_schema_name** [string] This parameter is deprecated

* **partition_table_name** [string] This parameter is deprecated

* **partition_table_partition_column_min_name** [string] This parameter is deprecated

* **partition_table_partition_column_max_name** [string] This parameter is deprecated

* last_modified_column : string

* **mysql_catalog_matches_schema** [boolean] This attribute is no longer available; defaults to true but cannot be used.

* **chunking_method** [string] This parameter is deprecated

* first_row_is_header : boolean

* **export_action** [string] The kind of export action you want to have the export execute. Set to "newsprsht" if you want a new worksheet inside a new spreadsheet. Set to "newwksht" if you want a new worksheet inside an existing spreadsheet. Set to "updatewksht" if you want to overwrite an existing worksheet inside an existing spreadsheet. Set to "appendwksht" if you want to append to the end of an existing worksheet inside an existing spreadsheet. Default is set to "newsprsht"

* **sql_query** [string] If you are doing a Google Sheet export, this is your SQL query.

* contact_lists : string

* soql_query : string

* include_deleted_records : boolean

- state : string
- created_at : string/date-time
- updated_at : string/date-time
- **last_run** [dict::]

    – id : integer

    – state : string

    – **created_at** [string/time] The time that the run was queued.

    – **started_at** [string/time] The time that the run started.

    – **finished_at** [string/time] The time that the run completed.

    – **error** [string] The error message for this run, if present.

- **user** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this import.
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

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**
>
> [*civis.response.Response*](#)
>
> - **name** [string] The name of the import.
> - **sync_type** [string] The type of sync to perform; one of Dbsync, AutoImport, GdocImport, GdocExport, and Salesforce.
> - **source** [dict::]
>
>   - remote_host_id : integer
>
>   - credential_id : integer
>
>   - **additional_credentials** [list] Array that holds additional credentials used for specific imports. For salesforce imports, the first and only element is the client credential id. For DB Syncs, the first element is an SSL private key credential id, and the second element is the corresponding public key credential id.
>
>   - name : string
>
> - **destination** [dict::]
>
>   - remote_host_id : integer
>
>   - credential_id : integer
>
>   - **additional_credentials** [list] Array that holds additional credentials used for specific imports. For salesforce imports, the first and only element is the client credential id. For DB Syncs, the first element is an SSL private key credential id, and the second element is the corresponding public key credential id.
>
>   - name : string
>
> - **schedule** [dict::]
>
>   - **scheduled** [boolean] If the item is scheduled.

- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

- **scheduled_hours** [list] Hours of the day it is scheduled on.

- **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

  - **urls** [list] URLs to receive a POST request at job completion

  - **success_email_subject** [string] Custom subject line for success e-mail.

  - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

  - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

  - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

  - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

  - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

  - **success_on** [boolean] If success email notifications are on.

  - **failure_on** [boolean] If failure email notifications are on.

- **parent_id** [integer] Parent id to trigger this import from
- **id** [integer] The ID for the import.
- is_outbound : boolean
- **job_type** [string] The job type of this import.
- **syncs** [list::] List of syncs. - id : integer - source : dict:

```
- id : integer
    The ID of the table or file, if available.
- path : string
    The path of the dataset to sync from; for a
→database source,
    schema.tablename. If you are doing a Google
→Sheet export, this
    can be blank. This is a legacy parameter, it
→is recommended you
    use one of the following: databaseTable, file,
→googleWorksheet,
    salesforce
```

(continues on next page)

```
- database_table : dict::
    - schema : string
        The database schema name.
    - table : string
        The database table name.
    - use_without_schema : boolean
        This attribute is no longer available;␣
→defaults to false
        but cannot be used.
- file : dict::
    - id : integer
        The file id.
- google_worksheet : dict::
    - spreadsheet : string
        The spreadsheet document name.
    - spreadsheet_id : string
        The spreadsheet document id.
    - worksheet : string
        The worksheet tab name.
    - worksheet_id : integer
        The worksheet tab id.
- salesforce : dict::
    - object_name : string
        The Salesforce object name.
```

– **destination** [dict::]

* **path** [string] The schema.tablename to sync to. If you are doing a Google Sheet export, this is the spreadsheet and sheet name separated by a period. i.e. if you have a spreadsheet named "MySpreadsheet" and a sheet called "Sheet1" this field would be "MySpreadsheet.Sheet1". This is a legacy parameter, it is recommended you use one of the following: databaseTable, googleWorksheet

* **database_table** [dict::]

· **schema** [string] The database schema name.

· **table** [string] The database table name.

· **use_without_schema** [boolean] This attribute is no longer available; defaults to false but cannot be used.

* **google_worksheet** [dict::]

· **spreadsheet** [string] The spreadsheet document name.

· **spreadsheet_id** [string] The spreadsheet document id.

· **worksheet** [string] The worksheet tab name.

· **worksheet_id** [integer] The worksheet tab id.

– **advanced_options** [dict::]

* max_errors : integer

* existing_table_rows : string

* diststyle : string

* distkey : string

* sortkey1 : string

* sortkey2 : string

* column_delimiter : string

* **column_overrides** [dict] Hash used for overriding auto-detected names and types, with keys being the index of the column being overridden.

* **escaped** [boolean] If true, escape quotes with a backslash; otherwise, escape quotes by double-quoting. Defaults to false.

* identity_column : string

* row_chunk_size : integer

* wipe_destination_table : boolean

* truncate_long_lines : boolean

* invalid_char_replacement : string

* verify_table_row_counts : boolean

* **partition_column_name** [string] This parameter is deprecated

* **partition_schema_name** [string] This parameter is deprecated

* **partition_table_name** [string] This parameter is deprecated

* **partition_table_partition_column_min_name** [string] This parameter is deprecated

* **partition_table_partition_column_max_name** [string] This parameter is deprecated

* last_modified_column : string

* **mysql_catalog_matches_schema** [boolean] This attribute is no longer available; defaults to true but cannot be used.

* **chunking_method** [string] This parameter is deprecated

* first_row_is_header : boolean

> > * **export_action** [string] The kind of export action
> >   you want to have the export execute. Set to
> >   "newsprsht" if you want a new worksheet in-
> >   side a new spreadsheet. Set to "newwksht" if
> >   you want a new worksheet inside an existing
> >   spreadsheet. Set to "updatewksht" if you want
> >   to overwrite an existing worksheet inside an ex-
> >   isting spreadsheet. Set to "appendwksht" if you
> >   want to append to the end of an existing work-
> >   sheet inside an existing spreadsheet. Default is
> >   set to "newsprsht"
> >
> > * **sql_query** [string] If you are doing a Google Sheet
> >   export, this is your SQL query.
> >
> > * contact_lists : string
> >
> > * soql_query : string
> >
> > * include_deleted_records : boolean

- state : string
- created_at : string/date-time
- updated_at : string/date-time
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

- **user** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this import.
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).

> • **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

put_files_csv(*id*, *source*, *destination*, *first_row_is_header*, *, *name='DEFAULT'*, *column_delimiter='DEFAULT'*, *escaped='DEFAULT'*, *compression='DEFAULT'*, *existing_table_rows='DEFAULT'*, *max_errors='DEFAULT'*, *table_columns='DEFAULT'*, *loosen_types='DEFAULT'*, *execution='DEFAULT'*, *redshift_destination_options='DEFAULT'*)

> Replace all attributes of this CSV Import
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the import.
> > >
> > > **source** [dict::]
> > >
> > > > • **file_ids** [list] The file ID(s) to import, if importing Civis file(s).
> > > > • **storage_path** [dict::]
> > > >
> > > > > – **storage_host_id** [integer] The ID of the source storage host.
> > > > >
> > > > > – **credential_id** [integer] The ID of the credentials for the source storage host.
> > > > >
> > > > > – **file_paths** [list] The file or directory path(s) within the bucket from which to import. E.g. the file_path for "s3://mybucket/files/all/" would be "/files/all/"If specifying a directory path, the job will import every file found under that path. All files must have the same column layout and file format (e.g., compression, columnDelimiter, etc.).
> > >
> > > **destination** [dict::]
> > >
> > > > • **schema** [string] The destination schema name.
> > > > • **table** [string] The destination table name.
> > > > • **remote_host_id** [integer] The ID of the destination database host.
> > > > • **credential_id** [integer] The ID of the credentials for the destination database.
> > > > • **primary_keys** [list] A list of column(s) which together uniquely identify a row in the destination table.These columns must not contain NULL values. If the import mode is "upsert", this field is required;see the Civis Helpdesk article on "Advanced CSV Imports via the Civis API" for more information.
> > > > • **last_modified_keys** [list] A list of the columns indicating a record has been updated.If the destination table does not exist, and the import mode is "upsert", this field is required.
> > >
> > > **first_row_is_header** [boolean] A boolean value indicating whether or not the first row of the source file is a header row.
> > >
> > > **name** [string, optional] The name of the import.
> > >
> > > **column_delimiter** [string, optional] The column delimiter for the file. Valid arguments are "comma", "tab", and "pipe". Defaults to "comma".
> > >
> > > **escaped** [boolean, optional] A boolean value indicating whether or not the source file has quotes escaped with a backslash.Defaults to false.
> > >
> > > **compression** [string, optional] The type of compression of the source file. Valid arguments are "gzip" and "none". Defaults to "none".
> > >
> > > **existing_table_rows** [string, optional] The behavior if a destination table with the requested name already exists. One of "fail", "truncate", "append", "drop", or "upsert".Defaults to "fail".
> > >
> > > **max_errors** [integer, optional] The maximum number of rows with errors to ignore before failing. This option is not supported for Postgres databases.
> > >
> > > **table_columns** [list, optional::] An array of hashes corresponding to the columns in

the order they appear in the source file. Each hash should have keys for database column "name" and "sqlType".This parameter is required if the table does not exist, the table is being dropped, or the columns in the source file do not appear in the same order as in the destination table.The "sqlType" key is not required when appending to an existing table. - name : string

The column name.

- **sql_type** [string] The SQL type of the column.

loosen_types [boolean, optional] If true, SQL types with precisions/lengths will have these values increased to accommodate data growth in future loads. Type loosening only occurs on table creation. Defaults to false.

execution [string, optional] In upsert mode, controls the movement of data in upsert mode. If set to "delayed", the data will be moved after a brief delay. If set to "immediate", the data will be moved immediately. In non-upsert modes, controls the speed at which detailed column stats appear in the data catalogue. Defaults to "delayed", to accommodate concurrent upserts to the same table and speedier non-upsert imports.

redshift_destination_options [dict, optional::]
- **diststyle** [string] The diststyle to use for the table. One of "even", "all", or "key".
- **distkey** [string] Distkey for this table in Redshift
- **sortkeys** [list] Sortkeys for this table in Redshift. Please provide a maximum of two.

Returns

civis.response.Response
- **id** [integer] The ID for the import.
- **name** [string] The name of the import.
- **source** [dict::]

    – **file_ids** [list] The file ID(s) to import, if importing Civis file(s).

    – **storage_path** [dict::]

        * **storage_host_id** [integer] The ID of the source storage host.

        * **credential_id** [integer] The ID of the credentials for the source storage host.

        * **file_paths** [list] The file or directory path(s) within the bucket from which to import. E.g. the file_path for "s3://mybucket/files/all/" would be "/files/all/"If specifying a directory path, the job will import every file found under that path. All files must have the same column layout and file format (e.g., compression, columnDelimiter, etc.).

- **destination** [dict::]

    – **schema** [string] The destination schema name.

    – **table** [string] The destination table name.

    – **remote_host_id** [integer] The ID of the destination database host.

    – **credential_id** [integer] The ID of the credentials for the destination database.

- **primary_keys** [list] A list of column(s) which together
  uniquely identify a row in the destination table.These
  columns must not contain NULL values. If the im-
  port mode is "upsert", this field is required;see the Civis
  Helpdesk article on "Advanced CSV Imports via the
  Civis API" for more information.

- **last_modified_keys** [list] A list of the columns indicating a
  record has been updated.If the destination table does not
  exist, and the import mode is "upsert", this field is re-
  quired.

- **first_row_is_header** [boolean] A boolean value indicating whether or not
  the first row of the source file is a header row.
- **column_delimiter** [string] The column delimiter for the file. Valid argu-
  ments are "comma", "tab", and "pipe". Defaults to "comma".
- **escaped** [boolean] A boolean value indicating whether or not the source
  file has quotes escaped with a backslash.Defaults to false.
- **compression** [string] The type of compression of the source file. Valid
  arguments are "gzip" and "none". Defaults to "none".
- **existing_table_rows** [string] The behavior if a destination table with the
  requested name already exists. One of "fail", "truncate", "append",
  "drop", or "upsert".Defaults to "fail".
- **max_errors** [integer] The maximum number of rows with errors to ignore
  before failing. This option is not supported for Postgres databases.
- **table_columns** [list::] An array of hashes corresponding to the columns
  in the order they appear in the source file. Each hash should have
  keys for database column "name" and "sqlType".This parameter is
  required if the table does not exist, the table is being dropped, or the
  columns in the source file do not appear in the same order as in the
  destination table.The "sqlType" key is not required when appending
  to an existing table. - name : string

    The column name.

    - **sql_type** [string] The SQL type of the column.

- **loosen_types** [boolean] If true, SQL types with precisions/lengths will
  have these values increased to accommodate data growth in future
  loads. Type loosening only occurs on table creation. Defaults to
  false.
- **execution** [string] In upsert mode, controls the movement of data in up-
  sert mode. If set to "delayed", the data will be moved after a brief
  delay. If set to "immediate", the data will be moved immediately.
  In non-upsert modes, controls the speed at which detailed column
  stats appear in the data catalogue. Defaults to "delayed", to accom-
  modate concurrent upserts to the same table and speedier non-upsert
  imports.
- **redshift_destination_options** [dict::]

    - **diststyle** [string] The diststyle to use for the table. One of
      "even", "all", or "key".

    - **distkey** [string] Distkey for this table in Redshift

    - **sortkeys** [list] Sortkeys for this table in Redshift. Please pro-
      vide a maximum of two.
- **hidden** [boolean] The hidden status of the item.

> - **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

put_files_csv_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**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > - **id** [integer] The ID for the import.
> > > > - **name** [string] The name of the import.
> > > > - **source** [dict::]
> > > >
> > > > > – **file_ids** [list] The file ID(s) to import, if importing Civis file(s).
> > > > >
> > > > > – **storage_path** [dict::]
> > > > >
> > > > > > ∗ **storage_host_id** [integer] The ID of the source storage host.
> > > > > >
> > > > > > ∗ **credential_id** [integer] The ID of the credentials for the source storage host.
> > > > > >
> > > > > > ∗ **file_paths** [list] The file or directory path(s) within the bucket from which to import. E.g. the file_path for "s3://mybucket/files/all/" would be "/files/all/"If specifying a directory path, the job will import every file found under that path. All files must have the same column layout and file format (e.g., compression, columnDelimiter, etc.).
> > > >
> > > > - **destination** [dict::]
> > > >
> > > > > – **schema** [string] The destination schema name.
> > > > >
> > > > > – **table** [string] The destination table name.
> > > > >
> > > > > – **remote_host_id** [integer] The ID of the destination database host.
> > > > >
> > > > > – **credential_id** [integer] The ID of the credentials for the destination database.
> > > > >
> > > > > – **primary_keys** [list] A list of column(s) which together uniquely identify a row in the destination table.These columns must not contain NULL values. If the import mode is "upsert", this field is required;see the Civis Helpdesk article on "Advanced CSV Imports via the Civis API" for more information.
> > > > >
> > > > > – **last_modified_keys** [list] A list of the columns indicating a record has been updated.If the destination table does not exist, and the import mode is "upsert", this field is required.
> > > >
> > > > - **first_row_is_header** [boolean] A boolean value indicating whether or not the first row of the source file is a header row.
> > > > - **column_delimiter** [string] The column delimiter for the file. Valid arguments are "comma", "tab", and "pipe". Defaults to "comma".

- **escaped** [boolean] A boolean value indicating whether or not the source file has quotes escaped with a backslash.Defaults to false.
- **compression** [string] The type of compression of the source file. Valid arguments are "gzip" and "none". Defaults to "none".
- **existing_table_rows** [string] The behavior if a destination table with the requested name already exists. One of "fail", "truncate", "append", "drop", or "upsert".Defaults to "fail".
- **max_errors** [integer] The maximum number of rows with errors to ignore before failing. This option is not supported for Postgres databases.
- **table_columns** [list::] An array of hashes corresponding to the columns in the order they appear in the source file. Each hash should have keys for database column "name" and "sqlType".This parameter is required if the table does not exist, the table is being dropped, or the columns in the source file do not appear in the same order as in the destination table.The "sqlType" key is not required when appending to an existing table. - name : string

  The column name.

    – **sql_type** [string] The SQL type of the column.

- **loosen_types** [boolean] If true, SQL types with precisions/lengths will have these values increased to accommodate data growth in future loads. Type loosening only occurs on table creation. Defaults to false.
- **execution** [string] In upsert mode, controls the movement of data in upsert mode. If set to "delayed", the data will be moved after a brief delay. If set to "immediate", the data will be moved immediately. In non-upsert modes, controls the speed at which detailed column stats appear in the data catalogue. Defaults to "delayed", to accommodate concurrent upserts to the same table and speedier non-upsert imports.
- **redshift_destination_options** [dict::]

    – **diststyle** [string] The diststyle to use for the table. One of "even", "all", or "key".

    – **distkey** [string] Distkey for this table in Redshift

    – **sortkeys** [list] Sortkeys for this table in Redshift. Please provide a maximum of two.
- **hidden** [boolean] The hidden status of the item.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

put_projects(*id*, *project_id*)

    Add an Import to a project

        Parameters

            id [integer] The ID of the Import.

            project_id [integer] The ID of the project.

        Returns

            None Response code 204: success

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

    Set the permissions groups has on this object

        Parameters

> **id** [integer] The ID of the resource that is shared.
> **group_ids** [list] An array of one or more group IDs.
> **permission_level** [string] Options are: "read", "write", or "manage".
> **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> **send_shared_email** [boolean, optional] Send email to the recipients of a share.

Returns
> [*civis.response.Response*](#)
>> • **readers** [dict::]
>>> – **users** [list::]
>>>> ∗ id : integer
>>>>
>>>> ∗ name : string
>>>
>>> – **groups** [list::]
>>>> ∗ id : integer
>>>>
>>>> ∗ name : string
>>
>> • **writers** [dict::]
>>> – **users** [list::]
>>>> ∗ id : integer
>>>>
>>>> ∗ name : string
>>>
>>> – **groups** [list::]
>>>> ∗ id : integer
>>>>
>>>> ∗ name : string
>>
>> • **owners** [dict::]
>>> – **users** [list::]
>>>> ∗ id : integer
>>>>
>>>> ∗ name : string
>>>
>>> – **groups** [list::]
>>>> ∗ id : integer
>>>>
>>>> ∗ name : string
>>
>> • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
>> • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_shares_users(*id*, *user_ids*, *permission_level*, *, *share_email_body='DEFAULT'*, *send_shared_email='DEFAULT'*)
> Set the permissions users have on this object

Parameters
> **id** [integer] The ID of the resource that is shared.
> **user_ids** [list] An array of one or more user IDs.
> **permission_level** [string] Options are: "read", "write", or "manage".
> **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> **send_shared_email** [boolean, optional] Send email to the recipients of a share.

Returns
> [*civis.response.Response*](#)
>> • **readers** [dict::]

> > > > – **users** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > > – **groups** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > • **writers** [dict::]

> > > > – **users** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > > – **groups** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > • **owners** [dict::]

> > > > – **users** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > > – **groups** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_syncs(*id*, *sync_id*, *source*, *destination*, ***, *advanced_options='DEFAULT'*)

> Update a sync

> > **Parameters**

> > > **id** [integer] The ID of the import to fetch.

> > > **sync_id** [integer] The ID of the sync to fetch.

> > > **source** [dict::]

> > > > • **path** [string] The path of the dataset to sync from; for a database source, schema.tablename. If you are doing a Google Sheet export, this can be blank. This is a legacy parameter, it is recommended you use one of the following: databaseTable, file, googleWorksheet, salesforce
> > > > • **database_table** [dict::]

> > > > > – **schema** [string] The database schema name.

> > > > > – **table** [string] The database table name.

> > > > > – **use_without_schema** [boolean] This attribute is no longer available; defaults to false but cannot be used.

> > > > • file : dict
> > > > • **google_worksheet** [dict::]

> > > > > – **spreadsheet** [string] The spreadsheet document name.

- **spreadsheet_id** [string] The spreadsheet document id.

- **worksheet** [string] The worksheet tab name.

- **worksheet_id** [integer] The worksheet tab id.

- **salesforce** [dict::]

  - **object_name** [string] The Salesforce object name.

destination [dict::]

- **path** [string] The schema.tablename to sync to. If you are doing a Google Sheet export, this is the spreadsheet and sheet name separated by a period. i.e. if you have a spreadsheet named "MySpreadsheet" and a sheet called "Sheet1" this field would be "MySpreadsheet.Sheet1". This is a legacy parameter, it is recommended you use one of the following: databaseTable, googleWorksheet

- **database_table** [dict::]

  - **schema** [string] The database schema name.

  - **table** [string] The database table name.

  - **use_without_schema** [boolean] This attribute is no longer available; defaults to false but cannot be used.

- **google_worksheet** [dict::]

  - **spreadsheet** [string] The spreadsheet document name.

  - **spreadsheet_id** [string] The spreadsheet document id.

  - **worksheet** [string] The worksheet tab name.

  - **worksheet_id** [integer] The worksheet tab id.

advanced_options [dict, optional::]

- max_errors : integer
- existing_table_rows : string
- diststyle : string
- distkey : string
- sortkey1 : string
- sortkey2 : string
- column_delimiter : string
- **column_overrides** [dict] Hash used for overriding auto-detected names and types, with keys being the index of the column being overridden.
- **escaped** [boolean] If true, escape quotes with a backslash; otherwise, escape quotes by double-quoting. Defaults to false.
- identity_column : string
- row_chunk_size : integer
- wipe_destination_table : boolean
- truncate_long_lines : boolean
- invalid_char_replacement : string
- verify_table_row_counts : boolean
- **partition_column_name** [string] This parameter is deprecated
- **partition_schema_name** [string] This parameter is deprecated
- **partition_table_name** [string] This parameter is deprecated
- **partition_table_partition_column_min_name** [string] This parameter is deprecated
- **partition_table_partition_column_max_name** [string] This parameter is deprecated
- last_modified_column : string

- **mysql_catalog_matches_schema** [boolean] This attribute is no longer available; defaults to true but cannot be used.
- **chunking_method** [string] This parameter is deprecated
- first_row_is_header : boolean
- **export_action** [string] The kind of export action you want to have the export execute. Set to "newsprsht" if you want a new worksheet inside a new spreadsheet. Set to "newwksht" if you want a new worksheet inside an existing spreadsheet. Set to "updatewksht" if you want to overwrite an existing worksheet inside an existing spreadsheet. Set to "appendwksht" if you want to append to the end of an existing worksheet inside an existing spreadsheet. Default is set to "newsprsht"
- **sql_query** [string] If you are doing a Google Sheet export, this is your SQL query.
- contact_lists : string
- soql_query : string
- include_deleted_records : boolean

Returns

> [*civis.response.Response*](#)
>
> - id : integer
> - **source** [dict::]
>
>> – **id** [integer] The ID of the table or file, if available.
>>
>> – **path** [string] The path of the dataset to sync from; for a database source, schema.tablename. If you are doing a Google Sheet export, this can be blank. This is a legacy parameter, it is recommended you use one of the following: databaseTable, file, googleWorksheet, salesforce
>>
>> – **database_table** [dict::]
>>
>>> * **schema** [string] The database schema name.
>>>
>>> * **table** [string] The database table name.
>>>
>>> * **use_without_schema** [boolean] This attribute is no longer available; defaults to false but cannot be used.
>>
>> – **file** [dict::]
>>
>>> * **id** [integer] The file id.
>>
>> – **google_worksheet** [dict::]
>>
>>> * **spreadsheet** [string] The spreadsheet document name.
>>>
>>> * **spreadsheet_id** [string] The spreadsheet document id.
>>>
>>> * **worksheet** [string] The worksheet tab name.
>>>
>>> * **worksheet_id** [integer] The worksheet tab id.
>>
>> – **salesforce** [dict::]
>>
>>> * **object_name** [string] The Salesforce object name.
>
> - **destination** [dict::]

- **path** [string] The schema.tablename to sync to. If you are
  doing a Google Sheet export, this is the spreadsheet and
  sheet name separated by a period. i.e. if you have a
  spreadsheet named "MySpreadsheet" and a sheet called
  "Sheet1" this field would be "MySpreadsheet.Sheet1".
  This is a legacy parameter, it is recommended you use
  one of the following: databaseTable, googleWorksheet

- **database_table** [dict::]

  - * **schema** [string] The database schema name.

  - * **table** [string] The database table name.

  - * **use_without_schema** [boolean] This attribute is
    no longer available; defaults to false but cannot
    be used.

- **google_worksheet** [dict::]

  - * **spreadsheet** [string] The spreadsheet document
    name.

  - * **spreadsheet_id** [string] The spreadsheet docu-
    ment id.

  - * **worksheet** [string] The worksheet tab name.

  - * **worksheet_id** [integer] The worksheet tab id.

- **advanced_options** [dict::]

  - max_errors : integer

  - existing_table_rows : string

  - diststyle : string

  - distkey : string

  - sortkey1 : string

  - sortkey2 : string

  - column_delimiter : string

  - **column_overrides** [dict] Hash used for overriding auto-
    detected names and types, with keys being the index of
    the column being overridden.

  - **escaped** [boolean] If true, escape quotes with a backslash;
    otherwise, escape quotes by double-quoting. Defaults
    to false.

  - identity_column : string

  - row_chunk_size : integer

  - wipe_destination_table : boolean

  - truncate_long_lines : boolean

  - invalid_char_replacement : string

  - verify_table_row_counts : boolean

- **partition_column_name** [string] This parameter is deprecated

- **partition_schema_name** [string] This parameter is deprecated

- **partition_table_name** [string] This parameter is deprecated

- **partition_table_partition_column_min_name** [string] This parameter is deprecated

- **partition_table_partition_column_max_name** [string] This parameter is deprecated

- last_modified_column : string

- **mysql_catalog_matches_schema** [boolean] This attribute is no longer available; defaults to true but cannot be used.

- **chunking_method** [string] This parameter is deprecated

- first_row_is_header : boolean

- **export_action** [string] The kind of export action you want to have the export execute. Set to "newsprsht" if you want a new worksheet inside a new spreadsheet. Set to "newwksht" if you want a new worksheet inside an existing spreadsheet. Set to "updatewksht" if you want to overwrite an existing worksheet inside an existing spreadsheet. Set to "appendwksht" if you want to append to the end of an existing worksheet inside an existing spreadsheet. Default is set to "newsprsht"

- **sql_query** [string] If you are doing a Google Sheet export, this is your SQL query.

- contact_lists : string

- soql_query : string

- include_deleted_records : boolean

put_syncs_archive(*id*, *sync_id*, ***, *status='DEFAULT'*)

> Update the archive status of this sync
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the import to fetch.
> > >
> > > **sync_id** [integer] The ID of the sync to fetch.
> > >
> > > **status** [boolean, optional] The desired archived status of the sync.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - id : integer
> > > > - **source** [dict::]
> > > >
> > > > > - **id** [integer] The ID of the table or file, if available.
> > > > >
> > > > > - **path** [string] The path of the dataset to sync from; for a database source, schema.tablename. If you are doing a Google Sheet export, this can be blank. This is a legacy parameter, it is recommended you use one of the following: databaseTable, file, googleWorksheet, salesforce
> > > > >
> > > > > - **database_table** [dict::]

             ∗ **schema** [string] The database schema name.

             ∗ **table** [string] The database table name.

             ∗ **use_without_schema** [boolean] This attribute is no longer available; defaults to false but cannot be used.

      – **file** [dict::]

             ∗ **id** [integer] The file id.

      – **google_worksheet** [dict::]

             ∗ **spreadsheet** [string] The spreadsheet document name.

             ∗ **spreadsheet_id** [string] The spreadsheet document id.

             ∗ **worksheet** [string] The worksheet tab name.

             ∗ **worksheet_id** [integer] The worksheet tab id.

      – **salesforce** [dict::]

             ∗ **object_name** [string] The Salesforce object name.

- **destination** [dict::]

      – **path** [string] The schema.tablename to sync to. If you are doing a Google Sheet export, this is the spreadsheet and sheet name separated by a period. i.e. if you have a spreadsheet named "MySpreadsheet" and a sheet called "Sheet1" this field would be "MySpreadsheet.Sheet1". This is a legacy parameter, it is recommended you use one of the following: databaseTable, googleWorksheet

      – **database_table** [dict::]

             ∗ **schema** [string] The database schema name.

             ∗ **table** [string] The database table name.

             ∗ **use_without_schema** [boolean] This attribute is no longer available; defaults to false but cannot be used.

      – **google_worksheet** [dict::]

             ∗ **spreadsheet** [string] The spreadsheet document name.

             ∗ **spreadsheet_id** [string] The spreadsheet document id.

             ∗ **worksheet** [string] The worksheet tab name.

             ∗ **worksheet_id** [integer] The worksheet tab id.

- **advanced_options** [dict::]

      – max_errors : integer

      – existing_table_rows : string

      – diststyle : string

- – distkey : string

- – sortkey1 : string

- – sortkey2 : string

- – column_delimiter : string

- – **column_overrides** [dict] Hash used for overriding auto-detected names and types, with keys being the index of the column being overridden.

- – **escaped** [boolean] If true, escape quotes with a backslash; otherwise, escape quotes by double-quoting. Defaults to false.

- – identity_column : string

- – row_chunk_size : integer

- – wipe_destination_table : boolean

- – truncate_long_lines : boolean

- – invalid_char_replacement : string

- – verify_table_row_counts : boolean

- – **partition_column_name** [string] This parameter is deprecated

- – **partition_schema_name** [string] This parameter is deprecated

- – **partition_table_name** [string] This parameter is deprecated

- – **partition_table_partition_column_min_name** [string] This parameter is deprecated

- – **partition_table_partition_column_max_name** [string] This parameter is deprecated

- – last_modified_column : string

- – **mysql_catalog_matches_schema** [boolean] This attribute is no longer available; defaults to true but cannot be used.

- – **chunking_method** [string] This parameter is deprecated

- – first_row_is_header : boolean

- – **export_action** [string] The kind of export action you want to have the export execute. Set to "newsprsht" if you want a new worksheet inside a new spreadsheet. Set to "newwksht" if you want a new worksheet inside an existing spreadsheet. Set to "updatewksht" if you want to overwrite an existing worksheet inside an existing spreadsheet. Set to "appendwksht" if you want to append to the end of an existing worksheet inside an existing spreadsheet. Default is set to "newsprsht"

- – **sql_query** [string] If you are doing a Google Sheet export, this is your SQL query.

- – contact_lists : string

- soql_query : string

- include_deleted_records : boolean

put_transfer(*id*, *user_id*, *include_dependencies*, ***, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

Transfer ownership of this object to another user

Parameters

id [integer] The ID of the resource that is shared.

user_id [integer] ID of target user

include_dependencies [boolean] Whether or not to give manage permissions on all dependencies

email_body [string, optional] Custom body text for e-mail sent on transfer.

send_email [boolean, optional] Send email to the target user of the transfer?

Returns

civis.response.Response

- **dependencies** [list::] Dependent objects for this object - object_type : string

    Dependent object type

    - **fco_type** [string] Human readable dependent object type

    - **id** [integer] Dependent object ID

    - **name** [string] Dependent object name, or nil if the requesting user cannot read this object

    - **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user

    - **shared** [boolean] Whether dependent object was successfully shared with target user

Jobs

class Jobs(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.jobs.list(...)
```

Methods

delete_projects(id, project_id)	Remove a Job from a project
delete_runs(id, run_id)	Cancel a run
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
get_runs(id, run_id)	Check status of a job
list(*[, state, type, q, permission, ...])	List Jobs
list_children(id)	Show nested tree of children that this job triggers
list_dependencies(id, *[, user_id])	List dependent objects for this object
list_parents(id)	Show chain of parents as a list that this job triggers from
list_projects(id, *[, hidden])	List the projects a Job belongs to
list_runs(id, *[, limit, page_num, order, ...])	List runs for the given job
list_runs_logs(id, run_id, *[, last_id, limit])	Get the logs for a run
list_runs_outputs(id, run_id, *[, limit, ...])	List the outputs for a run
list_shares(id)	List users and groups permissioned on this object
list_workflows(id, *[, archived])	List the workflows a job belongs to
post_runs(id)	Run a job
post_trigger_email(id)	Generate and retrieve trigger email address
put_archive(id, status)	Update the archive status of this object
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, ...[, ...])	Set the permissions users have on this object
put_transfer(id, user_id, ...[, email_body, ...])	Transfer ownership of this object to another user

——

delete_projects(*id*, *project_id*)

Remove a Job from a project

> **Parameters**
>> **id** [integer] The ID of the Job.
>> **project_id** [integer] The ID of the project.
>
> **Returns**
>> **None** Response code 204: success

delete_runs(*id*, *run_id*)

Cancel a run

> **Parameters**
>> **id** [integer] The ID of the Job.
>> **run_id** [integer] The ID of the Run.
>
> **Returns**
>> **None** Response code 202: success

delete_shares_groups(*id*, *group_id*)

Revoke the permissions a group has on this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>> **group_id** [integer] The 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] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] The 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**
> >
> > > *civis.response.Response*
> >
> > > > • id : integer
> > > > • name : string
> > > > • type : string
> > > > • from_template_id : integer
> > > > • **state** [string] Whether the job is idle, queued, running, cancelled, or failed.
> > > > • created_at : string/date-time
> > > > • updated_at : string/date-time
> > > > • **runs** [list::] Information about the most recent runs of the job. - id : integer - state : string - created_at : string/time
> > > >
> > > > > The time that the run was queued.
> > > >
> > > > > > – **started_at** [string/time] The time that the run started.
> > > > > >
> > > > > > – **finished_at** [string/time] The time that the run completed.
> > > > > >
> > > > > > – **error** [string] The error message for this run, if present.
> > > >
> > > > • **last_run** [dict::]
> > > >
> > > > > > – id : integer
> > > > > >
> > > > > > – state : string
> > > > > >
> > > > > > – **created_at** [string/time] The time that the run was queued.
> > > > > >
> > > > > > – **started_at** [string/time] The time that the run started.
> > > > > >
> > > > > > – **finished_at** [string/time] The time that the run completed.
> > > > > >
> > > > > > – **error** [string] The error message for this run, if present.
> > > >
> > > > • **hidden** [boolean] The hidden status of the item.
> > > > • **archived** [string] The archival status of the requested item(s).
> > > > • **author** [dict::]
> > > >
> > > > > > – **id** [integer] The ID of this user.
> > > > > >
> > > > > > – **name** [string] This user's name.
> > > > > >
> > > > > > – **username** [string] This user's username.
> > > > > >
> > > > > > – **initials** [string] This user's initials.
> > > > > >
> > > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > • **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
> > > > • success_email_subject : string
> > > > • success_email_body : string

- running_as_user : string
- run_by_user : string
- **schedule** [dict::]

  – **scheduled** [boolean] If the item is scheduled.

  – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  – **scheduled_hours** [list] Hours of the day it is scheduled on.

  – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

  – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

  – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

`get_runs`(*id*, *run_id*)

> Check status of a job
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the Job.
> > >
> > > **run_id** [integer] The ID of the Run.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > - id : integer
> > > - state : string
> > > - **created_at** [string/time] The time that the run was queued.
> > > - **started_at** [string/time] The time that the run started.
> > > - **finished_at** [string/time] The time that the run completed.
> > > - **error** [string] The error message for this run, if present.

`list`(***, *state='DEFAULT'*, *type='DEFAULT'*, *q='DEFAULT'*, *permission='DEFAULT'*, *scheduled='DEFAULT'*, *hidden='DEFAULT'*, *archived='DEFAULT'*, *author='DEFAULT'*, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)

> List Jobs
>
> > **Parameters**
> >
> > > **state** [string, optional] The job's state. One or more of queued, running, succeeded, failed, and cancelled. Specify multiple values as a comma-separated list (e.g., "A,B").
> > >
> > > **type** [string, optional] The job's type. Specify multiple values as a comma-separated list (e.g., "A,B").
> > >
> > > **q** [string, optional] Query string to search on the id, name, and job type.
> > >
> > > **permission** [string, optional] A permissions string, one of "read", "write", or "manage". Lists only jobs for which the current user has that permission.
> > >
> > > **scheduled** [boolean, optional] If the item is scheduled.
> > >
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> > >
> > > **archived** [string, optional] The archival status of the requested item(s).
> > >
> > > **author** [string, optional] If specified, return items from any of these authors. It accepts a comma- separated list of user IDs.
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to its maximum of 50.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.

order [string, optional] The field on which to order the result set. Defaults to updated_at. Must be one of: updated_at.

order_dir [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

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

Returns

civis.response.PaginatedResponse

- id : integer
- name : string
- type : string
- from_template_id : integer
- **state** [string] Whether the job is idle, queued, running, cancelled, or failed.
- created_at : string/date-time
- updated_at : string/date-time
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **archived** [string] The archival status of the requested item(s).
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

list_children(*id*)

Show nested tree of children that this job triggers

Parameters

> **id** [integer] The ID for this job.
>
> **Returns**
>
> [*civis.response.Response*](#)
>
> > • id : integer
> > • name : string
> > • type : string
> > • from_template_id : integer
> > • state : string
> > • created_at : string/date-time
> > • updated_at : string/date-time
> > • **runs** [list::]
> >
> > > – id : integer
> > >
> > > – state : string
> > >
> > > – **created_at** [string/time] The time that the run was queued.
> > >
> > > – **started_at** [string/time] The time that the run started.
> > >
> > > – **finished_at** [string/time] The time that the run completed.
> > >
> > > – **error** [string] The error message for this run, if present.
> >
> > • **last_run** [dict::]
> >
> > > – id : integer
> > >
> > > – state : string
> > >
> > > – **created_at** [string/time] The time that the run was queued.
> > >
> > > – **started_at** [string/time] The time that the run started.
> > >
> > > – **finished_at** [string/time] The time that the run completed.
> > >
> > > – **error** [string] The error message for this run, if present.
> >
> > • children : list

list_dependencies(*id*, ***, *user_id='DEFAULT'*)

> List dependent objects for this object
>
> **Parameters**
>
> > **id** [integer] The ID of the resource that is shared.
> > **user_id** [integer, optional] ID of target user
>
> **Returns**
>
> [*civis.response.Response*](#)
>
> > • **object_type** [string] Dependent object type
> > • **fco_type** [string] Human readable dependent object type
> > • **id** [integer] Dependent object ID
> > • **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > • **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > • **shareable** [boolean] Whether or not the requesting user can share this object.

list_parents(*id*)

> Show chain of parents as a list that this job triggers from
>
> **Parameters**
>
> > **id** [integer] The ID for this job.
>
> **Returns**
>
> [*civis.response.Response*](#)

- id : integer
- name : string
- type : string
- from_template_id : integer
- **state** [string] Whether the job is idle, queued, running, cancelled, or failed.
- created_at : string/date-time
- updated_at : string/date-time
- **runs** [list::] Information about the most recent runs of the job. - id : integer
    - state : string - created_at : string/time

        The time that the run was queued.

        – **started_at** [string/time] The time that the run started.

        – **finished_at** [string/time] The time that the run completed.

        – **error** [string] The error message for this run, if present.

- **last_run** [dict::]

        – id : integer

        – state : string

        – **created_at** [string/time] The time that the run was queued.

        – **started_at** [string/time] The time that the run started.

        – **finished_at** [string/time] The time that the run completed.

        – **error** [string] The error message for this run, if present.
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **author** [dict::]

        – **id** [integer] The ID of this user.

        – **name** [string] This user's name.

        – **username** [string] This user's username.

        – **initials** [string] This user's initials.

        – **online** [boolean] Whether this user is online.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- success_email_subject : string
- success_email_body : string
- running_as_user : string
- run_by_user : string
- **schedule** [dict::]

        – **scheduled** [boolean] If the item is scheduled.

        – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

        – **scheduled_hours** [list] Hours of the day it is scheduled on.

        – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

> > > – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
> > >
> > > – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

> **list_projects**(*id*, ***, *hidden='DEFAULT'*)
>
> > List the projects a Job belongs to
> >
> > > **Parameters**
> > >
> > > > **id** [integer] The ID of the Job.
> > > >
> > > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> > >
> > > **Returns**
> > >
> > > > [*civis.response.Response*](#)
> > > >
> > > > > • **id** [integer] The ID for this project.
> > > > > • **author** [dict::]
> > > > >
> > > > > > – **id** [integer] The ID of this user.
> > > > > >
> > > > > > – **name** [string] This user's name.
> > > > > >
> > > > > > – **username** [string] This user's username.
> > > > > >
> > > > > > – **initials** [string] This user's initials.
> > > > > >
> > > > > > – **online** [boolean] Whether this user is online.
> > > > >
> > > > > • **name** [string] The name of this project.
> > > > > • **description** [string] A description of the project.
> > > > > • **users** [list::] Users who can see the project. - id : integer
> > > > >
> > > > > > The ID of this user.
> > > > > >
> > > > > > – **name** [string] This user's name.
> > > > > >
> > > > > > – **username** [string] This user's username.
> > > > > >
> > > > > > – **initials** [string] This user's initials.
> > > > > >
> > > > > > – **online** [boolean] Whether this user is online.
> > > > >
> > > > > • auto_share : boolean
> > > > > • created_at : string/time
> > > > > • updated_at : string/time
> > > > > • **archived** [string] The archival status of the requested item(s).

> **list_runs**(*id*, ***, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)
>
> > List runs for the given job
> >
> > > **Parameters**
> > >
> > > > **id** [integer] The ID for this job.
> > > >
> > > > **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**
>> [*civis.response.PaginatedResponse*](#)
>>> - id : integer
>>> - state : string
>>> - **created_at** [string/time] The time that the run was queued.
>>> - **started_at** [string/time] The time that the run started.
>>> - **finished_at** [string/time] The time that the run completed.
>>> - **error** [string] The error message for this run, if present.

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

> Get the logs for a run
>> **Parameters**
>>> **id** [integer] The ID of the job.
>>> **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**
>>> [*civis.response.Response*](#)
>>>> - **id** [integer] The ID of the log.
>>>> - **created_at** [string/date-time] The time the log was created.
>>>> - **message** [string] The log message.
>>>> - **level** [string] The level of the log. One of un-
>>>>   known,fatal,error,warn,info,debug.

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

> List the outputs for a run
>> **Parameters**
>>> **id** [integer] The ID of the job.
>>> **run_id** [integer] The ID of the run.
>>> **limit** [integer, optional] Number of results to return. Defaults to its maximum of 50.
>>> **page_num** [integer, optional] Page number of the results to return. Defaults to the first
>>> page, 1.
>>> **order** [string, optional] The field on which to order the result set. Defaults to cre-
>>> ated_at. Must be one of: created_at, id.
>>> **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc
>>> (descending) defaulting to desc.
>>> **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use
>>> when more results than the maximum allowed by limit are needed. When True,
>>> limit and page_num are ignored. Defaults to False.

>> **Returns**
>>> [*civis.response.PaginatedResponse*](#)
>>>> - **object_type** [string] The type of the output. Valid values are File, Table,
>>>>   Report, Project, Credential, or JSONValue
>>>> - **object_id** [integer] The ID of the output.
>>>> - **name** [string] The name of the output.
>>>> - **link** [string] The hypermedia link to the output.
>>>> - value : string

list_shares(*id*)

List users and groups permissioned on this object

> **Parameters**
>
> > **id** [integer] The ID of the resource that is shared.
>
> **Returns**
>
> > *civis.response.Response*
> >
> > > - **readers** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > > - **writers** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > > - **owners** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_workflows(*id*, ***, *archived='DEFAULT'*)

List the workflows a job belongs to

> **Parameters**
>
> > **id** [integer]
> >
> > **archived** [string, optional] The archival status of the requested item(s).
>
> **Returns**
>
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID for this workflow.
> > > - **name** [string] The name of this workflow.
> > > - **description** [string] A description of the workflow.
> > > - **valid** [boolean] The validity of the workflow definition.
> > > - **file_id** [string] The file id for the s3 file containing the workflow configuration.

- **user** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **state** [string] The state of the workflow. State is "running" if any execution is running, otherwise reflects most recent execution state.
- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **allow_concurrent_executions** [boolean] Whether the workflow can execute when already running.
- **time_zone** [string] The time zone of this workflow.
- **next_execution_at** [string/time] The time of the next scheduled execution.
- **archived** [string] The archival status of the requested item(s).
- created_at : string/time
- updated_at : string/time

post_runs(*id*)

    Run a job

        Parameters

            id [integer] The ID for this job.

        Returns

        civis.response.Response

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

post_trigger_email(*id*)

    Generate and retrieve trigger email address

        Parameters

            id [integer] The ID for this job.

        Returns

        civis.response.Response

- **trigger_email** [string] Email address which may be used to trigger this job to run.

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

        civis.response.Response

- id : integer
- name : string
- type : string
- from_template_id : integer
- **state** [string] Whether the job is idle, queued, running, cancelled, or failed.
- created_at : string/date-time
- updated_at : string/date-time
- **runs** [list::] Information about the most recent runs of the job. - id : integer
  - state : string - created_at : string/time

    The time that the run was queued.

    – **started_at** [string/time] The time that the run started.

    – **finished_at** [string/time] The time that the run completed.

    – **error** [string] The error message for this run, if present.

- **last_run** [dict::]

    – id : integer

    – state : string

    – **created_at** [string/time] The time that the run was queued.

    – **started_at** [string/time] The time that the run started.

    – **finished_at** [string/time] The time that the run completed.

    – **error** [string] The error message for this run, if present.
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- **my_permission_level** [string] Your permission level on the object. One
    of "read", "write", or "manage".
- success_email_subject : string
- success_email_body : string
- running_as_user : string
- run_by_user : string
- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

– **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

– **scheduled_hours** [list] Hours of the day it is scheduled on.

– **scheduled_minutes** [list] Minutes of the day it is scheduled on.

– **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

– **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

put_projects(*id*, *project_id*)

> Add a Job to a project
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the Job.
> > >
> > > **project_id** [integer] The ID of the project.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

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

> Set the permissions groups has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **group_ids** [list] An array of one or more group IDs.
> > >
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > >
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > >
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **readers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **writers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **owners** [dict::]
> > > >
> > > > > – **users** [list::]

       ∗ id : integer

       ∗ name : string

      – **groups** [list::]

       ∗ id : integer

       ∗ name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

  Set the permissions users have on this object

   Parameters

    id [integer] The ID of the resource that is shared.

    user_ids [list] An array of one or more user IDs.

    permission_level [string] Options are: "read", "write", or "manage".

    share_email_body [string, optional] Custom body text for e-mail sent on a share.

    send_shared_email [boolean, optional] Send email to the recipients of a share.

   Returns

   [*civis.response.Response*](#)

- **readers** [dict::]

   – **users** [list::]

     ∗ id : integer

     ∗ name : string

   – **groups** [list::]

     ∗ id : integer

     ∗ name : string

- **writers** [dict::]

   – **users** [list::]

     ∗ id : integer

     ∗ name : string

   – **groups** [list::]

     ∗ id : integer

     ∗ name : string

- **owners** [dict::]

   – **users** [list::]

     ∗ id : integer

     ∗ name : string

   – **groups** [list::]

     ∗ id : integer

     ∗ name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_transfer(*id*, *user_id*, *include_dependencies*, *, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

Transfer ownership of this object to another user

Parameters

id [integer] The ID of the resource that is shared.

user_id [integer] ID of target user

include_dependencies [boolean] Whether or not to give manage permissions on all dependencies

email_body [string, optional] Custom body text for e-mail sent on transfer.

send_email [boolean, optional] Send email to the target user of the transfer?

Returns

civis.response.Response

- **dependencies** [list::] Dependent objects for this object - object_type : string

    Dependent object type

    - **fco_type** [string] Human readable dependent object type

    - **id** [integer] Dependent object ID

    - **name** [string] Dependent object name, or nil if the requesting user cannot read this object

    - **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user

    - **shared** [boolean] Whether dependent object was successfully shared with target user

Json_Values

class Json_Values(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.json_values.post(...)
```

Methods

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 JSON Value
list_dependencies(id, *[, user_id])	List dependent objects for this object
list_shares(id)	List users and groups permissioned on this object
patch(id, *[, name, value_str])	Update some attributes of this JSON Value
post(value_str, *[, name])	Create a JSON Value
put_shares_groups(id, group_ids, ...[, ...])	Set the permissions groups has on this object
put_shares_users(id, user_ids, ...[, ...])	Set the permissions users have on this object
put_transfer(id, user_id, ...[, email_body, ...])	Transfer ownership of this object to another user

—

delete_shares_groups(*id*, *group_id*)
    Revoke the permissions a group has on this object
        Parameters
            id [integer] The ID of the resource that is shared.
            group_id [integer] The 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] The ID of the resource that is shared.
            user_id [integer] The ID of the user.
        Returns
            None Response code 204: success

get(*id*)
    Get details about a JSON Value
        Parameters
            id [integer] The ID of the JSON Value.
        Returns
        civis.response.Response
            • **id** [integer] The ID of the JSON Value.
            • **name** [string] The name of the JSON Value.
            • **value** [string] The deserialized JSON value.

list_dependencies(*id*, *, *user_id='DEFAULT'*)
    List dependent objects for this object
        Parameters
            id [integer] The ID of the resource that is shared.
            user_id [integer, optional] ID of target user
        Returns
        civis.response.Response
            • **object_type** [string] Dependent object type
            • **fco_type** [string] Human readable dependent object type
            • **id** [integer] Dependent object ID
            • **name** [string] Dependent object name, or nil if the requesting user cannot read this object

- **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
- **shareable** [boolean] Whether or not the requesting user can share this object.

list_shares(*id*)

List users and groups permissioned on this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>
> **Returns**
>> [*civis.response.Response*](#)
>>> - **readers** [dict::]
>>>> – **users** [list::]
>>>>> * id : integer
>>>>>
>>>>> * name : string
>>>>
>>>> – **groups** [list::]
>>>>> * id : integer
>>>>>
>>>>> * name : string
>>>
>>> - **writers** [dict::]
>>>> – **users** [list::]
>>>>> * id : integer
>>>>>
>>>>> * name : string
>>>>
>>>> – **groups** [list::]
>>>>> * id : integer
>>>>>
>>>>> * name : string
>>>
>>> - **owners** [dict::]
>>>> – **users** [list::]
>>>>> * id : integer
>>>>>
>>>>> * name : string
>>>>
>>>> – **groups** [list::]
>>>>> * id : integer
>>>>>
>>>>> * name : string
>>>
>>> - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
>>> - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch(*id*, ***, *name='DEFAULT'*, *value_str='DEFAULT'*)

Update some attributes of this JSON Value

> **Parameters**
>> **id** [integer] The ID of the JSON Value.
>>
>> **name** [string, optional] The name of the JSON Value.
>>
>> **value_str** [string, optional] The JSON value to store. Should be a serialized JSON string. Limited to 1000000 bytes.
>
> **Returns**
>> [*civis.response.Response*](#)

- **id** [integer] The ID of the JSON Value.
- **name** [string] The name of the JSON Value.
- **value** [string] The deserialized JSON value.

post(*value_str*, *, *name='DEFAULT'*)

Create a JSON Value

> **Parameters**
>
> > **value_str** [string] The JSON value to store. Should be a serialized JSON string. Limited to 1000000 bytes.
> >
> > **name** [string, optional] The name of the JSON Value.
>
> **Returns**
>
> > [*civis.response.Response*](#)
> >
> > > - **id** [integer] The ID of the JSON Value.
> > > - **name** [string] The name of the JSON Value.
> > > - **value** [string] The deserialized JSON value.

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

Set the permissions groups has on this object

> **Parameters**
>
> > **id** [integer] The ID of the resource that is shared.
> >
> > **group_ids** [list] An array of one or more group IDs.
> >
> > **permission_level** [string] Options are: "read", "write", or "manage".
> >
> > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> >
> > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
>
> **Returns**
>
> > [*civis.response.Response*](#)
> >
> > > - **readers** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > > - **writers** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > > - **owners** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]

> > > > > * id : integer
> > > >
> > > > * name : string
> > >
> > > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **user_ids** [list] An array of one or more user IDs.
> > >
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > >
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > >
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > - **readers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > >
> > > > - **writers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > >
> > > > - **owners** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > >
> > > > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_transfer(*id*, *user_id*, *include_dependencies*, ***, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

> Transfer ownership of this object to another user
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] ID of target user
> > >
> > > **include_dependencies** [boolean] Whether or not to give manage permissions on all dependencies
> > >
> > > **email_body** [string, optional] Custom body text for e-mail sent on transfer.
> > >
> > > **send_email** [boolean, optional] Send email to the target user of the transfer?
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **dependencies** [list::] Dependent objects for this object - object_type : string
> > > >
> > > > > Dependent object type
> > > > >
> > > > > – **fco_type** [string] Human readable dependent object type
> > > > >
> > > > > – **id** [integer] Dependent object ID
> > > > >
> > > > > – **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > > > >
> > > > > – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > > > >
> > > > > – **shared** [boolean] Whether dependent object was successfully shared with target user

Match_Targets

class **Match_Targets**(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.match_targets.list_shares(...)
```

Methods

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 Match Target info
list()	List match targets
list_shares(id)	List users and groups permissioned on this object
patch(id, *[, name, target_file_name, archived])	Update a match target
post(name, *[, target_file_name, archived])	Create a new match target
put_archive(id, status)	Update the archive status of this object

—

delete_shares_groups(*id*, *group_id*)

> Revoke the permissions a group has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **group_id** [integer] The 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] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] The ID of the user.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

get(*id*)

> Show Match Target info
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the match target
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **id** [integer] The ID of the match target
> > > > • **name** [string] The name of the match target
> > > > • **target_file_name** [string] The name of the target file
> > > > • created_at : string/time
> > > > • updated_at : string/time
> > > > • **archived** [boolean] Whether the match target has been archived.

list()

> List match targets
>
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **id** [integer] The ID of the match target
> > > > • **name** [string] The name of the match target
> > > > • **target_file_name** [string] The name of the target file
> > > > • created_at : string/time
> > > > • updated_at : string/time
> > > > • **archived** [boolean] Whether the match target has been archived.

list_shares(*id*)

> List users and groups permissioned on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **readers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer

> > > > > ∗ name : string
>
> > > > – **groups** [list::]
> >
> > > > > ∗ id : integer
> >
> > > > > ∗ name : string
>
> > > • **writers** [dict::]
> >
> > > > – **users** [list::]
> >
> > > > > ∗ id : integer
> >
> > > > > ∗ name : string
> >
> > > > – **groups** [list::]
> >
> > > > > ∗ id : integer
> >
> > > > > ∗ name : string
>
> > > • **owners** [dict::]
> >
> > > > – **users** [list::]
> >
> > > > > ∗ id : integer
> >
> > > > > ∗ name : string
> >
> > > > – **groups** [list::]
> >
> > > > > ∗ id : integer
> >
> > > > > ∗ name : string
>
> > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch(*id*, *, *name='DEFAULT'*, *target_file_name='DEFAULT'*, *archived='DEFAULT'*)

> Update a match target
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the match target
> > >
> > > **name** [string, optional] The name of the match target
> > >
> > > **target_file_name** [string, optional] The name of the target file
> > >
> > > **archived** [boolean, optional] Whether the match target has been archived.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **id** [integer] The ID of the match target
> > > > • **name** [string] The name of the match target
> > > > • **target_file_name** [string] The name of the target file
> > > > • created_at : string/time
> > > > • updated_at : string/time
> > > > • **archived** [boolean] Whether the match target has been archived.

post(*name*, *, *target_file_name='DEFAULT'*, *archived='DEFAULT'*)

> Create a new match target
>
> > **Parameters**
> >
> > > **name** [string] The name of the match target
> > >
> > > **target_file_name** [string, optional] The name of the target file
> > >
> > > **archived** [boolean, optional] Whether the match target has been archived.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)

- **id** [integer] The ID of the match target
- **name** [string] The name of the match target
- **target_file_name** [string] The name of the target file
- created_at : string/time
- updated_at : string/time
- **archived** [boolean] Whether the match target has been archived.

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**
>> *civis.response.Response*
>>> - **id** [integer] The ID of the match target
>>> - **name** [string] The name of the match target
>>> - **target_file_name** [string] The name of the target file
>>> - created_at : string/time
>>> - updated_at : string/time
>>> - **archived** [boolean] Whether the match target has been archived.

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

Set the permissions groups has on this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>>
>> **group_ids** [list] An array of one or more group IDs.
>>
>> **permission_level** [string] Options are: "read", "write", or "manage".
>>
>> **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
>>
>> **send_shared_email** [boolean, optional] Send email to the recipients of a share.
>
> **Returns**
>> *civis.response.Response*
>>> - **readers** [dict::]
>>>> - **users** [list::]
>>>>> * id : integer
>>>>>
>>>>> * name : string
>>>>
>>>> - **groups** [list::]
>>>>> * id : integer
>>>>>
>>>>> * name : string
>>>
>>> - **writers** [dict::]
>>>> - **users** [list::]
>>>>> * id : integer
>>>>>
>>>>> * name : string
>>>>
>>>> - **groups** [list::]
>>>>> * id : integer
>>>>>
>>>>> * name : string
>>>
>>> - **owners** [dict::]
>>>> - **users** [list::]

> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **user_ids** [list] An array of one or more user IDs.
> > >
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > >
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > >
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > [*civis.response.Response*](#)
> >
> > > - **readers** [dict::]
> > >
> > > > – **users** [list::]
> > > >
> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string
> > > >
> > > > – **groups** [list::]
> > > >
> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string
> > >
> > > - **writers** [dict::]
> > >
> > > > – **users** [list::]
> > > >
> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string
> > > >
> > > > – **groups** [list::]
> > > >
> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string
> > >
> > > - **owners** [dict::]
> > >
> > > > – **users** [list::]
> > > >
> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string
> > > >
> > > > – **groups** [list::]
> > > >
> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string

> - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Media

class Media(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.media.list_spot_orders_shares(...)
```

Methods

delete_optimizations_runs(id, run_id)	Cancel a run
delete_optimizations_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_optimizations_shares_users(id, user_id)	Revoke the permissions a user has on this object
delete_ratecards_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_ratecards_shares_users(id, user_id)	Revoke the permissions a user has on this object
delete_spot_orders_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_spot_orders_shares_users(id, user_id)	Revoke the permissions a user has on this object
get_optimizations(id)	Show a single optimization
get_optimizations_runs(id, run_id)	Check status of a run
get_ratecards(id)	Get a Ratecard
get_spot_orders(id)	Show a single spot order
list_dmas(*[, name, number])	List all Designated Market Areas
list_optimizations(*[, archived, limit, ...])	List all optimizations
list_optimizations_runs(id, *[, limit, ...])	List runs for the given optimization
list_optimizations_runs_logs(id, run_id, *)	Get the logs for a run
list_optimizations_shares(id)	List users and groups permissioned on this object
list_ratecards(*[, archived, filename, ...])	List all ratecards
list_ratecards_shares(id)	List users and groups permissioned on this object
list_spot_orders(*[, id, archived])	List all spot orders
list_spot_orders_shares(id)	List users and groups permissioned on this object
list_targets(*[, name, identifier, data_source])	List all Media Targets
patch_optimizations(id, *[, name, runs, ...])	Edit an existing optimization
patch_ratecards(id, *[, filename, start_on, ...])	Update some attributes of this Ratecard
post_optimizations(runs, *[, name, ...])	Create a new optimization
post_optimizations_clone(id)	Clone an existing optimization

Table 39 – continued from previous page

post_optimizations_runs(id)	Start a run
post_ratecards(filename, start_on, end_on, ...)	Create a Ratecard
post_spot_orders(*[, body])	Create a spot order
put_optimizations_archive(id, status)	Update the archive status of this object
put_optimizations_shares_groups(id, ...[, ...])	Set the permissions groups has on this object
put_optimizations_shares_users(id, user_ids, ...)	Set the permissions users have on this object
put_ratecards(id, filename, start_on, ...)	Replace all attributes of this Ratecard
put_ratecards_archive(id, status)	Update the archive status of this object
put_ratecards_shares_groups(id, group_ids, ...)	Set the permissions groups has on this object
put_ratecards_shares_users(id, user_ids, ...)	Set the permissions users have on this object
put_spot_orders(id, *[, body])	Edit the specified spot order
put_spot_orders_archive(id, status)	Update the archive status of this object
put_spot_orders_shares_groups(id, group_ids, ...)	Set the permissions groups has on this object
put_spot_orders_shares_users(id, user_ids, ...)	Set the permissions users have on this object

—

delete_optimizations_runs(*id*, *run_id*)
>   Cancel a run
>>   **Parameters**
>>>   **id** [integer] The ID of the optimization.
>>>   **run_id** [integer] The ID of the run.
>>   **Returns**
>>>   **None** Response code 202: success

delete_optimizations_shares_groups(*id*, *group_id*)
>   Revoke the permissions a group has on this object
>>   **Parameters**
>>>   **id** [integer] The ID of the resource that is shared.
>>>   **group_id** [integer] The ID of the group.
>>   **Returns**
>>>   **None** Response code 204: success

delete_optimizations_shares_users(*id*, *user_id*)
>   Revoke the permissions a user has on this object
>>   **Parameters**
>>>   **id** [integer] The ID of the resource that is shared.
>>>   **user_id** [integer] The ID of the user.
>>   **Returns**
>>>   **None** Response code 204: success

delete_ratecards_shares_groups(*id*, *group_id*)
>   Revoke the permissions a group has on this object
>>   **Parameters**
>>>   **id** [integer] The ID of the resource that is shared.
>>>   **group_id** [integer] The ID of the group.
>>   **Returns**
>>>   **None** Response code 204: success

delete_ratecards_shares_users(*id*, *user_id*)

    Revoke the permissions a user has on this object

        Parameters

            id [integer] The ID of the resource that is shared.

            user_id [integer] The ID of the user.

        Returns

            None Response code 204: success

delete_spot_orders_shares_groups(*id*, *group_id*)

    Revoke the permissions a group has on this object

        Parameters

            id [integer] The ID of the resource that is shared.

            group_id [integer] The ID of the group.

        Returns

            None Response code 204: success

delete_spot_orders_shares_users(*id*, *user_id*)

    Revoke the permissions a user has on this object

        Parameters

            id [integer] The ID of the resource that is shared.

            user_id [integer] The ID of the user.

        Returns

            None Response code 204: success

get_optimizations(*id*)

    Show a single optimization

        Parameters

            id [integer] The optimization ID.

        Returns

         civis.response.Response

            • **id** [integer] The optimization ID.

            • **author** [dict::]

                – **id** [integer] The ID of this user.

                – **name** [string] This user's name.

                – **username** [string] This user's username.

                – **initials** [string] This user's initials.

                – **online** [boolean] Whether this user is online.

            • **name** [string] The name of the optimization.

            • created_at : string/time

            • updated_at : string/time

            • **finished_at** [string/date-time] The end time of the last run.

            • **state** [string] The state of the last run.

            • **last_run_id** [integer] The ID of the last run.

            • **spot_order_id** [integer] The ID for the spot order produced by the optimization.

            • **archived** [string] The archival status of the requested item(s).

            • **report_link** [string] A link to the visual report for the optimization.

            • **spot_order_link** [string] A link to the json version of the spot order.

            • **file_links** [list] Links to the csv and xml versions of the spot order.

            • **runs** [list::] The runs of the optimization. - market_id : integer

                The market ID.

- **start_date** [string/date] The start date for the media run.

- **end_date** [string/date] The end date for the media run.

- **force_cpm** [boolean] Whether to force optimization to use CPM data even if partition data is available.

- **reach_alpha** [number/float] A tuning parameter used to adjust RF.

- **syscodes** [list] The syscodes for the media run.

- **rate_cards** [list] The ratecards for the media run.

- **constraints** [list::] The constraints for the media run. - targets : list

  The targets to constrain.

  * **budget** [number/float] The maximum budget for these targets.

  * **frequency** [number/float] The maximum frequency for these targets.

- **programs** [list] An array of programs that the Civis Media Optimizer either exclude or limit to.An error will be thrown if exclude_programs is not also set.
- **networks** [list] An array of networks that the Civis Media Optimizer either exclude or limit to.An error will be thrown if exclude_networks is not also set.
- **exclude_programs** [boolean] If Civis Media Optimizer should exclude the programs in the programs parameter.If this value is set to false, it will make the optimization limit itself to the programs supplied through the programs parameter.An error will be thrown if programs is not also set.
- **exclude_networks** [boolean] If Civis Media Optimizer should exclude the networks in the networks parameter.If this value is set to false, it will make the optimization limit itself to the networks supplied through the networks.An error will be thrown if networks is not also set.
- **time_slot_percentages** [dict] The maximum amount of the budget spent on that particular day of the week, daypart, or specific time slot for broadcast and cable.

get_optimizations_runs(*id*, *run_id*)

Check status of a run

Parameters

id [integer] The ID of the optimization.

run_id [integer] The ID of the run.

Returns

civis.response.Response

- **id** [integer] The ID of the run.
- **optimization_id** [integer] The ID of the optimization.
- **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **is_cancel_requested** [boolean] True if run cancel requested, else false.
- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.
- **error** [string] The error, if any, returned by the run.

get_ratecards(*id*)

> Get a Ratecard
>
> > **Parameters**
> >
> > > **id** [integer]
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **id** [integer] The ratecard ID.
> > > > - **filename** [string] Name of the ratecard file.
> > > > - **start_on** [string/date] First day to which the ratecard applies.
> > > > - **end_on** [string/date] Last day to which the ratecard applies.
> > > > - **dma_number** [integer] Number of the DMA associated with the ratecard.
> > > > - **archived** [string] The archival status of the requested item(s).

get_spot_orders(*id*)

> Show a single spot order
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the spot order.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **id** [integer] The ID for the spot order.
> > > > - **archived** [string] The archival status of the requested item(s).
> > > > - **csv_s3_uri** [string] S3 URI for the spot order CSV file.
> > > > - **json_s3_uri** [string] S3 URI for the spot order JSON file.
> > > > - **xml_archive_s3_uri** [string] S3 URI for the spot order XML archive.
> > > > - **last_transform_job_id** [integer] ID of the spot order transformation job.

list_dmas(***, *name='DEFAULT'*, *number='DEFAULT'*)

> List all Designated Market Areas
>
> > **Parameters**
> >
> > > **name** [string, optional] If specified, will be used to filter the DMAs returned. Substring matching is supported with "%" and "*" wildcards (e.g., "name=%region%" will return both "region1" and "my region").
> > >
> > > **number** [integer, optional] If specified, will be used to filter the DMAS by number.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **name** [string] Name for the DMA region.
> > > > - **number** [integer] Identifier number for a DMA.

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

> List all optimizations
>
> > **Parameters**
> >
> > > **archived** [string, optional] The archival status of the requested item(s).
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 50.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to created_at. Must be one of: created_at, author, name.
> > >
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > >
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> >
> > **Returns**

> > > *civis.response.PaginatedResponse*
> > > - **id** [integer] The optimization ID.
> > > - **author** [dict::]
> > >
> > > > - **id** [integer] The ID of this user.
> > > >
> > > > - **name** [string] This user's name.
> > > >
> > > > - **username** [string] This user's username.
> > > >
> > > > - **initials** [string] This user's initials.
> > > >
> > > > - **online** [boolean] Whether this user is online.
> > > - **name** [string] The name of the optimization.
> > > - created_at : string/time
> > > - updated_at : string/time
> > > - **finished_at** [string/date-time] The end time of the last run.
> > > - **state** [string] The state of the last run.
> > > - **last_run_id** [integer] The ID of the last run.
> > > - **spot_order_id** [integer] The ID for the spot order produced by the optimization.
> > > - **archived** [string] The archival status of the requested item(s).

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

> List runs for the given optimization
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the optimization.
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 100.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to id. Must be one of: id.
> > >
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > >
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> >
> > **Returns**
> >
> > *civis.response.PaginatedResponse*
> > > - **id** [integer] The ID of the run.
> > > - **optimization_id** [integer] The ID of the optimization.
> > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > - **started_at** [string/time] The time the last run started at.
> > > - **finished_at** [string/time] The time the last run completed.
> > > - **error** [string] The error, if any, returned by the run.

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

> Get the logs for a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the optimization.
> > >
> > > **run_id** [integer] The ID of the run.
> > >
> > > **last_id** [integer, optional] The ID of the last log message received. Log entries with this ID value or lower will be omitted.Logs are sorted by ID if this value is provided,

and are otherwise sorted by createdAt.

   limit [integer, optional] The maximum number of log messages to return. Default of 10000.

Returns

civis.response.Response

- **id** [integer] The ID of the log.
- **created_at** [string/date-time] The time the log was created.
- **message** [string] The log message.
- **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

list_optimizations_shares(*id*)

List users and groups permissioned on this object

   Parameters

   id [integer] The ID of the resource that is shared.

   Returns

civis.response.Response

- **readers** [dict::]

    – **users** [list::]

        * id : integer

        * name : string

    – **groups** [list::]

        * id : integer

        * name : string

- **writers** [dict::]

    – **users** [list::]

        * id : integer

        * name : string

    – **groups** [list::]

        * id : integer

        * name : string

- **owners** [dict::]

    – **users** [list::]

        * id : integer

        * name : string

    – **groups** [list::]

        * id : integer

        * name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_ratecards(***, *archived='DEFAULT'*, *filename='DEFAULT'*, *dma_number='DEFAULT'*)

List all ratecards

   Parameters

> **archived** [string, optional] The archival status of the requested item(s).
>
> **filename** [string, optional] If specified, will be used to filter the ratecards returned. Substring matching is supported with "%" and "*" wildcards (e.g., "filename=%ratecard%" will return both "ratecard 1" and "my ratecard").
>
> **dma_number** [integer, optional] If specified, will be used to filter the ratecards by DMA.
>
> **Returns**
>
> *civis.response.Response*
>
> > - **id** [integer] The ratecard ID.
> > - **filename** [string] Name of the ratecard file.
> > - **start_on** [string/date] First day to which the ratecard applies.
> > - **end_on** [string/date] Last day to which the ratecard applies.
> > - **dma_number** [integer] Number of the DMA associated with the ratecard.
> > - **archived** [string] The archival status of the requested item(s).

list_ratecards_shares(*id*)

> List users and groups permissioned on this object
>
> **Parameters**
>
> > **id** [integer] The ID of the resource that is shared.
>
> **Returns**
>
> *civis.response.Response*
>
> > - **readers** [dict::]
> >
> >   - **users** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> >   - **groups** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> > - **writers** [dict::]
> >
> >   - **users** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> >   - **groups** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> > - **owners** [dict::]
> >
> >   - **users** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> >   - **groups** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

> • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_spot_orders(*, *id='DEFAULT'*, *archived='DEFAULT'*)
> List all spot orders
>> **Parameters**
>>> **id** [integer, optional] The ID for the spot order.
>>> **archived** [string, optional] The archival status of the requested item(s).
>> **Returns**
>> [*civis.response.Response*](#)
>>> • **id** [integer] The ID for the spot order.
>>> • **archived** [string] The archival status of the requested item(s).

list_spot_orders_shares(*id*)
> List users and groups permissioned on this object
>> **Parameters**
>>> **id** [integer] The ID of the resource that is shared.
>> **Returns**
>> [*civis.response.Response*](#)
>>> • **readers** [dict::]
>>>> – **users** [list::]
>>>>> ∗ id : integer
>>>>> ∗ name : string
>>>> – **groups** [list::]
>>>>> ∗ id : integer
>>>>> ∗ name : string
>>> • **writers** [dict::]
>>>> – **users** [list::]
>>>>> ∗ id : integer
>>>>> ∗ name : string
>>>> – **groups** [list::]
>>>>> ∗ id : integer
>>>>> ∗ name : string
>>> • **owners** [dict::]
>>>> – **users** [list::]
>>>>> ∗ id : integer
>>>>> ∗ name : string
>>>> – **groups** [list::]
>>>>> ∗ id : integer
>>>>> ∗ name : string
>>> • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
>>> • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

---

list_targets(*, *name='DEFAULT'*, *identifier='DEFAULT'*, *data_source='DEFAULT'*)

> List all Media Targets
>
> > **Parameters**
> >
> > > **name** [string, optional] The name of the target.
> > >
> > > **identifier** [string, optional] A unique identifier for this target.
> > >
> > > **data_source** [string, optional] The source of viewership data for this target.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > • **name** [string] The name of the target.
> > >
> > > • **identifier** [string] A unique identifier for this target.
> > >
> > > • **data_source** [string] The source of viewership data for this target.

patch_optimizations(*id*, *, *name='DEFAULT'*, *runs='DEFAULT'*, *programs='DEFAULT'*, *networks='DEFAULT'*, *exclude_programs='DEFAULT'*, *exclude_networks='DEFAULT'*, *time_slot_percentages='DEFAULT'*)

> Edit an existing optimization
>
> > **Parameters**
> >
> > > **id** [integer] The optimization ID.
> > >
> > > **name** [string, optional] The name of the optimization.
> > >
> > > **runs** [list, optional::] The runs of the optimization. - market_id : integer
> > >
> > > > The market ID.
> > >
> > > > • **start_date** [string/date] The start date for the media run.
> > > >
> > > > • **end_date** [string/date] The end date for the media run.
> > > >
> > > > • **force_cpm** [boolean] Whether to force optimization to use CPM data even if partition data is available.
> > > >
> > > > • **reach_alpha** [number/float] A tuning parameter used to adjust RF.
> > > >
> > > > • **syscodes** [list] The syscodes for the media run.
> > > >
> > > > • **rate_cards** [list] The ratecards for the media run.
> > > >
> > > > • **constraints** [list::] The constraints for the media run. - targets : list
> > > >
> > > > > The targets to constrain.
> > > >
> > > > > – **budget** [number/float] The maximum budget for these targets.
> > > > >
> > > > > – **frequency** [number/float] The maximum frequency for these targets.
> > >
> > > **programs** [list, optional] An array of programs that the Civis Media Optimizer either exclude or limit to.An error will be thrown if exclude_programs is not also set.
> > >
> > > **networks** [list, optional] An array of networks that the Civis Media Optimizer either exclude or limit to.An error will be thrown if exclude_networks is not also set.
> > >
> > > **exclude_programs** [boolean, optional] If Civis Media Optimizer should exclude the programs in the programs parameter.If this value is set to false, it will make the optimization limit itself to the programs supplied through the programs parameter.An error will be thrown if programs is not also set.
> > >
> > > **exclude_networks** [boolean, optional] If Civis Media Optimizer should exclude the networks in the networks parameter.If this value is set to false, it will make the optimization limit itself to the networks supplied through the networks.An error will be thrown if networks is not also set.
> > >
> > > **time_slot_percentages** [dict, optional] The maximum amount of the budget spent on that particular day of the week, daypart, or specific time slot for broadcast and cable.
> >
> > **Returns**
> >
> > *civis.response.Response*

- **id** [integer] The optimization ID.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- **name** [string] The name of the optimization.
- created_at : string/time
- updated_at : string/time
- **finished_at** [string/date-time] The end time of the last run.
- **state** [string] The state of the last run.
- **last_run_id** [integer] The ID of the last run.
- **spot_order_id** [integer] The ID for the spot order produced by the optimization.
- **archived** [string] The archival status of the requested item(s).
- **report_link** [string] A link to the visual report for the optimization.
- **spot_order_link** [string] A link to the json version of the spot order.
- **file_links** [list] Links to the csv and xml versions of the spot order.
- **runs** [list::] The runs of the optimization. - market_id : integer

    The market ID.

    – **start_date** [string/date] The start date for the media run.

    – **end_date** [string/date] The end date for the media run.

    – **force_cpm** [boolean] Whether to force optimization to use CPM data even if partition data is available.

    – **reach_alpha** [number/float] A tuning parameter used to adjust RF.

    – **syscodes** [list] The syscodes for the media run.

    – **rate_cards** [list] The ratecards for the media run.

    – **constraints** [list::] The constraints for the media run. - targets : list

        The targets to constrain.

        ∗ **budget** [number/float] The maximum budget for these targets.

        ∗ **frequency** [number/float] The maximum frequency for these targets.
- **programs** [list] An array of programs that the Civis Media Optimizer either exclude or limit to.An error will be thrown if exclude_programs is not also set.
- **networks** [list] An array of networks that the Civis Media Optimizer either exclude or limit to.An error will be thrown if exclude_networks is not also set.
- **exclude_programs** [boolean] If Civis Media Optimizer should exclude the programs in the programs parameter.If this value is set to false,

it will make the optimization limit itself to the programs supplied through the programs parameter.An error will be thrown if programs is not also set.

- **exclude_networks** [boolean] If Civis Media Optimizer should exclude the networks in the networks parameter.If this value is set to false, it will make the optimization limit itself to the networks supplied through the networks.An error will be thrown if networks is not also set.
- **time_slot_percentages** [dict] The maximum amount of the budget spent on that particular day of the week, daypart, or specific time slot for broadcast and cable.

patch_ratecards(*id*, ***, *filename='DEFAULT'*, *start_on='DEFAULT'*, *end_on='DEFAULT'*, *dma_number='DEFAULT'*)

Update some attributes of this Ratecard

> **Parameters**
>> **id** [integer] The ratecard ID.
>> **filename** [string, optional] Name of the ratecard file.
>> **start_on** [string/date, optional] First day to which the ratecard applies.
>> **end_on** [string/date, optional] Last day to which the ratecard applies.
>> **dma_number** [integer, optional] Number of the DMA associated with the ratecard.
>
> **Returns**
>> [*civis.response.Response*](#)
>>> - **id** [integer] The ratecard ID.
>>> - **filename** [string] Name of the ratecard file.
>>> - **start_on** [string/date] First day to which the ratecard applies.
>>> - **end_on** [string/date] Last day to which the ratecard applies.
>>> - **dma_number** [integer] Number of the DMA associated with the ratecard.
>>> - **archived** [string] The archival status of the requested item(s).

post_optimizations(*runs*, ***, *name='DEFAULT'*, *programs='DEFAULT'*, *networks='DEFAULT'*, *exclude_programs='DEFAULT'*, *exclude_networks='DEFAULT'*, *time_slot_percentages='DEFAULT'*)

Create a new optimization

> **Parameters**
>> **runs** [list::] The runs of the optimization. - market_id : integer
>>> The market ID.
>>
>>> - **start_date** [string/date] The start date for the media run.
>>> - **end_date** [string/date] The end date for the media run.
>>> - **force_cpm** [boolean] Whether to force optimization to use CPM data even if partition data is available.
>>> - **reach_alpha** [number/float] A tuning parameter used to adjust RF.
>>> - **syscodes** [list] The syscodes for the media run.
>>> - **rate_cards** [list] The ratecards for the media run.
>>> - **constraints** [list::] The constraints for the media run. - targets : list
>>>
>>>> The targets to constrain.
>>>
>>>> – **budget** [number/float] The maximum budget for these targets.
>>>
>>>> – **frequency** [number/float] The maximum frequency for these targets.
>>
>> **name** [string, optional] The name of the optimization.
>> **programs** [list, optional] An array of programs that the Civis Media Optimizer either exclude or limit to.An error will be thrown if exclude_programs is not also set.

    networks [list, optional] An array of networks that the Civis Media Optimizer either exclude or limit to.An error will be thrown if exclude_networks is not also set.

    exclude_programs [boolean, optional] If Civis Media Optimizer should exclude the programs in the programs parameter.If this value is set to false, it will make the optimization limit itself to the programs supplied through the programs parameter.An error will be thrown if programs is not also set.

    exclude_networks [boolean, optional] If Civis Media Optimizer should exclude the networks in the networks parameter.If this value is set to false, it will make the optimization limit itself to the networks supplied through the networks.An error will be thrown if networks is not also set.

    time_slot_percentages [dict, optional] The maximum amount of the budget spent on that particular day of the week, daypart, or specific time slot for broadcast and cable.

Returns

    civis.response.Response

- **id** [integer] The optimization ID.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **name** [string] The name of the optimization.
- created_at : string/time
- updated_at : string/time
- **finished_at** [string/date-time] The end time of the last run.
- **state** [string] The state of the last run.
- **last_run_id** [integer] The ID of the last run.
- **spot_order_id** [integer] The ID for the spot order produced by the optimization.
- **archived** [string] The archival status of the requested item(s).
- **report_link** [string] A link to the visual report for the optimization.
- **spot_order_link** [string] A link to the json version of the spot order.
- **file_links** [list] Links to the csv and xml versions of the spot order.
- **runs** [list::] The runs of the optimization. - market_id : integer

    The market ID.

    - **start_date** [string/date] The start date for the media run.

    - **end_date** [string/date] The end date for the media run.

    - **force_cpm** [boolean] Whether to force optimization to use CPM data even if partition data is available.

    - **reach_alpha** [number/float] A tuning parameter used to adjust RF.

    - **syscodes** [list] The syscodes for the media run.

    - **rate_cards** [list] The ratecards for the media run.

    - **constraints** [list::] The constraints for the media run. - targets : list

The targets to constrain.

* **budget** [number/float] The maximum budget for these targets.

* **frequency** [number/float] The maximum frequency for these targets.

- **programs** [list] An array of programs that the Civis Media Optimizer either exclude or limit to.An error will be thrown if exclude_programs is not also set.
- **networks** [list] An array of networks that the Civis Media Optimizer either exclude or limit to.An error will be thrown if exclude_networks is not also set.
- **exclude_programs** [boolean] If Civis Media Optimizer should exclude the programs in the programs parameter.If this value is set to false, it will make the optimization limit itself to the programs supplied through the programs parameter.An error will be thrown if programs is not also set.
- **exclude_networks** [boolean] If Civis Media Optimizer should exclude the networks in the networks parameter.If this value is set to false, it will make the optimization limit itself to the networks supplied through the networks.An error will be thrown if networks is not also set.
- **time_slot_percentages** [dict] The maximum amount of the budget spent on that particular day of the week, daypart, or specific time slot for broadcast and cable.

post_optimizations_clone(*id*)

Clone an existing optimization

Parameters

id [integer] The optimization ID.

Returns

civis.response.Response

- **id** [integer] The optimization ID.
- **author** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.
- **name** [string] The name of the optimization.
- created_at : string/time
- updated_at : string/time
- **finished_at** [string/date-time] The end time of the last run.
- **state** [string] The state of the last run.
- **last_run_id** [integer] The ID of the last run.
- **spot_order_id** [integer] The ID for the spot order produced by the optimization.
- **archived** [string] The archival status of the requested item(s).
- **report_link** [string] A link to the visual report for the optimization.
- **spot_order_link** [string] A link to the json version of the spot order.
- **file_links** [list] Links to the csv and xml versions of the spot order.
- **runs** [list::] The runs of the optimization. - market_id : integer

The market ID.

- **start_date** [string/date] The start date for the media run.

- **end_date** [string/date] The end date for the media run.

- **force_cpm** [boolean] Whether to force optimization to use CPM data even if partition data is available.

- **reach_alpha** [number/float] A tuning parameter used to adjust RF.

- **syscodes** [list] The syscodes for the media run.

- **rate_cards** [list] The ratecards for the media run.

- **constraints** [list::] The constraints for the media run. - targets : list

  The targets to constrain.

  * **budget** [number/float] The maximum budget for these targets.

  * **frequency** [number/float] The maximum frequency for these targets.

- **programs** [list] An array of programs that the Civis Media Optimizer either exclude or limit to.An error will be thrown if exclude_programs is not also set.
- **networks** [list] An array of networks that the Civis Media Optimizer either exclude or limit to.An error will be thrown if exclude_networks is not also set.
- **exclude_programs** [boolean] If Civis Media Optimizer should exclude the programs in the programs parameter.If this value is set to false, it will make the optimization limit itself to the programs supplied through the programs parameter.An error will be thrown if programs is not also set.
- **exclude_networks** [boolean] If Civis Media Optimizer should exclude the networks in the networks parameter.If this value is set to false, it will make the optimization limit itself to the networks supplied through the networks.An error will be thrown if networks is not also set.
- **time_slot_percentages** [dict] The maximum amount of the budget spent on that particular day of the week, daypart, or specific time slot for broadcast and cable.

post_optimizations_runs(*id*)

Start a run

Parameters

id [integer] The ID of the optimization.

Returns

civis.response.Response

- **id** [integer] The ID of the run.
- **optimization_id** [integer] The ID of the optimization.
- **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **is_cancel_requested** [boolean] True if run cancel requested, else false.
- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.

- **error** [string] The error, if any, returned by the run.

post_ratecards(*filename*, *start_on*, *end_on*, *dma_number*)

> Create a Ratecard
>
> > **Parameters**
> >
> > > **filename** [string] Name of the ratecard file.
> > >
> > > **start_on** [string/date] First day to which the ratecard applies.
> > >
> > > **end_on** [string/date] Last day to which the ratecard applies.
> > >
> > > **dma_number** [integer] Number of the DMA associated with the ratecard.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **id** [integer] The ratecard ID.
> > > > - **filename** [string] Name of the ratecard file.
> > > > - **start_on** [string/date] First day to which the ratecard applies.
> > > > - **end_on** [string/date] Last day to which the ratecard applies.
> > > > - **dma_number** [integer] Number of the DMA associated with the ratecard.
> > > > - **archived** [string] The archival status of the requested item(s).

post_spot_orders(***, *body='DEFAULT'*)

> Create a spot order
>
> > **Parameters**
> >
> > > **body** [string, optional] CSV body of a spot order.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **id** [integer] The ID for the spot order.
> > > > - **archived** [string] The archival status of the requested item(s).
> > > > - **csv_s3_uri** [string] S3 URI for the spot order CSV file.
> > > > - **json_s3_uri** [string] S3 URI for the spot order JSON file.
> > > > - **xml_archive_s3_uri** [string] S3 URI for the spot order XML archive.
> > > > - **last_transform_job_id** [integer] ID of the spot order transformation job.

put_optimizations_archive(*id*, *status*)

> Update the archive status of this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the object.
> > >
> > > **status** [boolean] The desired archived status of the object.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **id** [integer] The optimization ID.
> > > > - **author** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > - **name** [string] The name of the optimization.
> > > > - created_at : string/time
> > > > - updated_at : string/time
> > > > - **finished_at** [string/date-time] The end time of the last run.
> > > > - **state** [string] The state of the last run.
> > > > - **last_run_id** [integer] The ID of the last run.
> > > > - **spot_order_id** [integer] The ID for the spot order produced by the optimization.

- **archived** [string] The archival status of the requested item(s).
- **report_link** [string] A link to the visual report for the optimization.
- **spot_order_link** [string] A link to the json version of the spot order.
- **file_links** [list] Links to the csv and xml versions of the spot order.
- **runs** [list::] The runs of the optimization. - market_id : integer

    The market ID.

    – **start_date** [string/date] The start date for the media run.

    – **end_date** [string/date] The end date for the media run.

    – **force_cpm** [boolean] Whether to force optimization to use CPM data even if partition data is available.

    – **reach_alpha** [number/float] A tuning parameter used to adjust RF.

    – **syscodes** [list] The syscodes for the media run.

    – **rate_cards** [list] The ratecards for the media run.

    – **constraints** [list::] The constraints for the media run. - targets : list

        The targets to constrain.

        ∗ **budget** [number/float] The maximum budget for these targets.

        ∗ **frequency** [number/float] The maximum frequency for these targets.

- **programs** [list] An array of programs that the Civis Media Optimizer either exclude or limit to.An error will be thrown if exclude_programs is not also set.
- **networks** [list] An array of networks that the Civis Media Optimizer either exclude or limit to.An error will be thrown if exclude_networks is not also set.
- **exclude_programs** [boolean] If Civis Media Optimizer should exclude the programs in the programs parameter.If this value is set to false, it will make the optimization limit itself to the programs supplied through the programs parameter.An error will be thrown if programs is not also set.
- **exclude_networks** [boolean] If Civis Media Optimizer should exclude the networks in the networks parameter.If this value is set to false, it will make the optimization limit itself to the networks supplied through the networks.An error will be thrown if networks is not also set.
- **time_slot_percentages** [dict] The maximum amount of the budget spent on that particular day of the week, daypart, or specific time slot for broadcast and cable.

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

Set the permissions groups has on this object

Parameters

id [integer] The ID of the resource that is shared.

group_ids [list] An array of one or more group IDs.

permission_level [string] Options are: "read", "write", or "manage".

> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > > - **readers** [dict::]
> > > > >
> > > > > > > - **users** [list::]
> > > > > > >
> > > > > > > > > * id : integer
> > > > > > > >
> > > > > > > > * name : string
> > > > > > >
> > > > > > > - **groups** [list::]
> > > > > > >
> > > > > > > > > * id : integer
> > > > > > > >
> > > > > > > > * name : string
> > > > >
> > > > > - **writers** [dict::]
> > > > >
> > > > > > > - **users** [list::]
> > > > > > >
> > > > > > > > > * id : integer
> > > > > > > >
> > > > > > > > * name : string
> > > > > > >
> > > > > > > - **groups** [list::]
> > > > > > >
> > > > > > > > > * id : integer
> > > > > > > >
> > > > > > > > * name : string
> > > > >
> > > > > - **owners** [dict::]
> > > > >
> > > > > > > - **users** [list::]
> > > > > > >
> > > > > > > > > * id : integer
> > > > > > > >
> > > > > > > > * name : string
> > > > > > >
> > > > > > > - **groups** [list::]
> > > > > > >
> > > > > > > > > * id : integer
> > > > > > > >
> > > > > > > > * name : string
> > > > >
> > > > > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_ids** [list] An array of one or more user IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > > - **readers** [dict::]
> > > > >
> > > > > > > - **users** [list::]
> > > > > > >
> > > > > > > > > * id : integer

> > > > > ∗ name : string
> > >
> > > – **groups** [list::]
> > >
> > > > > ∗ id : integer
> > > >
> > > > > ∗ name : string
> >
> > - **writers** [dict::]
> >
> > > – **users** [list::]
> > >
> > > > > ∗ id : integer
> > > >
> > > > > ∗ name : string
> > >
> > > – **groups** [list::]
> > >
> > > > > ∗ id : integer
> > > >
> > > > > ∗ name : string
> >
> > - **owners** [dict::]
> >
> > > – **users** [list::]
> > >
> > > > > ∗ id : integer
> > > >
> > > > > ∗ name : string
> > >
> > > – **groups** [list::]
> > >
> > > > > ∗ id : integer
> > > >
> > > > > ∗ name : string
> >
> > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_ratecards(*id*, *filename*, *start_on*, *end_on*, *dma_number*)

> Replace all attributes of this Ratecard
>
> > **Parameters**
> >
> > > **id** [integer] The ratecard ID.
> > >
> > > **filename** [string] Name of the ratecard file.
> > >
> > > **start_on** [string/date] First day to which the ratecard applies.
> > >
> > > **end_on** [string/date] Last day to which the ratecard applies.
> > >
> > > **dma_number** [integer] Number of the DMA associated with the ratecard.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > - **id** [integer] The ratecard ID.
> > > - **filename** [string] Name of the ratecard file.
> > > - **start_on** [string/date] First day to which the ratecard applies.
> > > - **end_on** [string/date] Last day to which the ratecard applies.
> > > - **dma_number** [integer] Number of the DMA associated with the ratecard.
> > > - **archived** [string] The archival status of the requested item(s).

put_ratecards_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**
> >
> > *civis.response.Response*

- **id** [integer] The ratecard ID.
- **filename** [string] Name of the ratecard file.
- **start_on** [string/date] First day to which the ratecard applies.
- **end_on** [string/date] Last day to which the ratecard applies.
- **dma_number** [integer] Number of the DMA associated with the ratecard.
- **archived** [string] The archival status of the requested item(s).

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

Set the permissions groups has on this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>> **group_ids** [list] An array of one or more group IDs.
>> **permission_level** [string] Options are: "read", "write", or "manage".
>> **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
>> **send_shared_email** [boolean, optional] Send email to the recipients of a share.
>
> **Returns**
>> *civis.response.Response*
>>> - **readers** [dict::]
>>>> – **users** [list::]
>>>>> * id : integer
>>>>> * name : string
>>>> – **groups** [list::]
>>>>> * id : integer
>>>>> * name : string
>>> - **writers** [dict::]
>>>> – **users** [list::]
>>>>> * id : integer
>>>>> * name : string
>>>> – **groups** [list::]
>>>>> * id : integer
>>>>> * name : string
>>> - **owners** [dict::]
>>>> – **users** [list::]
>>>>> * id : integer
>>>>> * name : string
>>>> – **groups** [list::]
>>>>> * id : integer
>>>>> * name : string
>>> - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
>>> - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

Set the permissions users have on this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>> **user_ids** [list] An array of one or more user IDs.
>> **permission_level** [string] Options are: "read", "write", or "manage".
>> **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
>> **send_shared_email** [boolean, optional] Send email to the recipients of a share.
>
> **Returns**
>> [*civis.response.Response*](#)
>>> • **readers** [dict::]
>>>> – **users** [list::]
>>>>> ∗ id : integer
>>>>>
>>>>> ∗ name : string
>>>>
>>>> – **groups** [list::]
>>>>> ∗ id : integer
>>>>>
>>>>> ∗ name : string
>>>
>>> • **writers** [dict::]
>>>> – **users** [list::]
>>>>> ∗ id : integer
>>>>>
>>>>> ∗ name : string
>>>>
>>>> – **groups** [list::]
>>>>> ∗ id : integer
>>>>>
>>>>> ∗ name : string
>>>
>>> • **owners** [dict::]
>>>> – **users** [list::]
>>>>> ∗ id : integer
>>>>>
>>>>> ∗ name : string
>>>>
>>>> – **groups** [list::]
>>>>> ∗ id : integer
>>>>>
>>>>> ∗ name : string
>>>
>>> • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
>>> • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_spot_orders(*id*, *, *body='DEFAULT'*)

Edit the specified spot order

> **Parameters**
>> **id** [integer] The ID for the spot order.
>> **body** [string, optional] CSV body of a spot order.
>
> **Returns**
>> [*civis.response.Response*](#)
>>> • **id** [integer] The ID for the spot order.

- **archived** [string] The archival status of the requested item(s).
- **csv_s3_uri** [string] S3 URI for the spot order CSV file.
- **json_s3_uri** [string] S3 URI for the spot order JSON file.
- **xml_archive_s3_uri** [string] S3 URI for the spot order XML archive.
- **last_transform_job_id** [integer] ID of the spot order transformation job.

put_spot_orders_archive(*id*, *status*)

> Update the archive status of this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the object.
> > >
> > > **status** [boolean] The desired archived status of the object.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID for the spot order.
> > > - **archived** [string] The archival status of the requested item(s).
> > > - **csv_s3_uri** [string] S3 URI for the spot order CSV file.
> > > - **json_s3_uri** [string] S3 URI for the spot order JSON file.
> > > - **xml_archive_s3_uri** [string] S3 URI for the spot order XML archive.
> > > - **last_transform_job_id** [integer] ID of the spot order transformation job.

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

> Set the permissions groups has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **group_ids** [list] An array of one or more group IDs.
> > >
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > >
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > >
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > - **readers** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > > - **writers** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > > - **owners** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer

> > > > > > ∗ name : string

> > > > – **groups** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object

> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_ids** [list] An array of one or more user IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.

> > **Returns**

> > [*civis.response.Response*](#)

> > > • **readers** [dict::]

> > > > – **users** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > > – **groups** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > • **writers** [dict::]

> > > > – **users** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > > – **groups** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > • **owners** [dict::]

> > > > – **users** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > > – **groups** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

> • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Models

class **Models**(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.models.list_types(...)
```

Methods

delete_builds(id, build_id)	Cancel a build
delete_projects(id, project_id)	Remove a Model 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(*[, model_name, training_table_name, ...])	List
list_builds(id, *[, limit, page_num, order, ...])	List builds for the given model
list_builds_logs(id, build_id, *[, last_id, ...])	Get the logs for a build
list_dependencies(id, *[, user_id])	List dependent objects for this object
list_projects(id, *[, hidden])	List the projects a Model 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
put_archive(id, status)	Update the archive status of this object
put_projects(id, project_id)	Add a Model to a project
put_shares_groups(id, group_ids, ...[, ...])	Set the permissions groups has on this object
put_shares_users(id, user_ids, ...[, ...])	Set the permissions users have on this object
put_transfer(id, user_id, ...[, email_body, ...])	Transfer ownership of this object to another user

——

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 Model from a project
>> **Parameters**
>>> **id** [integer] The ID of the Model.

> > **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] The ID of the resource that is shared.
> > > **group_id** [integer] The 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] The ID of the resource that is shared.
> > > **user_id** [integer] The ID of the user.
> > **Returns**
> > > **None** Response code 204: success

get(*id*)

> Retrieve model configuration
> > **Parameters**
> > > **id** [integer] The ID of the model.
> > **Returns**
> > > [*civis.response.Response*](#)
> > > > - **id** [integer] The ID of the model.
> > > > - **table_name** [string] The qualified name of the table containing the training set from which to build the model.
> > > > - **database_id** [integer] The ID of the database holding the training set table used to build the model.
> > > > - **credential_id** [integer] The ID of the credential used to read the target table. Defaults to the user's default credential.
> > > > - **model_name** [string] The name of the model.
> > > > - **description** [string] A description of the model.
> > > > - **interaction_terms** [boolean] Whether to search for interaction terms.
> > > > - **box_cox_transformation** [boolean] Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.
> > > > - **model_type_id** [integer] The ID of the model's type.
> > > > - **primary_key** [string] The unique ID (primary key) of the training dataset.
> > > > - **dependent_variable** [string] The dependent variable of the training dataset.
> > > > - **dependent_variable_order** [list] The order of dependent variables, especially useful for Ordinal Modeling.
> > > > - **excluded_columns** [list] A list of columns which will be considered ineligible to be independent variables.
> > > > - **limiting_sql** [string] A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., "id > 105").
> > > > - **active_build_id** [integer] The ID of the current active build, the build used to score predictions.
> > > > - **cross_validation_parameters** [dict] Cross validation parameter grid for tree methods, e.g. {"n_estimators": [100, 200, 500], "learning_rate": [0.01, 0.1], "max_depth": [2, 3]}.
> > > > - **number_of_folds** [integer] Number of folds for cross validation. Default value is 5.

- **notifications** [dict::]

  – **urls** [list] URLs to receive a POST request at job completion

  – **success_email_subject** [string] Custom subject line for success e-mail.

  – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

  – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

  – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

  – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

  – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

  – **success_on** [boolean] If success email notifications are on.

  – **failure_on** [boolean] If failure email notifications are on.

- **schedule** [dict::]

  – **scheduled** [boolean] If the item is scheduled.

  – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  – **scheduled_hours** [list] Hours of the day it is scheduled on.

  – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

  – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

  – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **parent_id** [integer] The ID of the parent job that will trigger this model.
- **running_as** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.

- **time_zone** [string] The time zone of this model.
- **last_run** [dict::]

  – id : integer

  – state : string

  – **created_at** [string/time] The time that the run was queued.

- **started_at** [string/time] The time that the run started.

- **finished_at** [string/time] The time that the run completed.

- **error** [string] The error message for this run, if present.
- **hidden** [boolean] The hidden status of the item.
- **user** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **created_at** [string/date-time] The time the model was created.
- **updated_at** [string/date-time] The time the model was updated.
- **current_build_state** [string] The status of the current model build. One of "succeeded", "failed", "queued", or "running,"or "idle", if no build has been attempted.
- **current_build_exception** [string] Exception message, if applicable, of the current model build.
- **builds** [list::] A list of trained models available for making predictions. - id : integer

    The ID of the model build.

    - **name** [string] The name of the model build.

    - **created_at** [string] The time the model build was created.

    - **description** [string] A description of the model build.

    - **root_mean_squared_error** [number/float] A key metric for continuous models. Nil for other model types.

    - **r_squared_error** [number/float] A key metric for continuous models. Nil for other model types.

    - **roc_auc** [number/float] A key metric for binary, multinomial, and ordinal models. Nil for other model types.

- **predictions** [list::] The tables upon which the model will be applied. - id : integer

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

    - **table_name** [string] The qualified name of the table on which to apply the predictive model.

    - **primary_key** [list] The primary key or composite keys of the table being predicted.

    - **limiting_sql** [string] A SQL WHERE clause used to scope the rows to be predicted.

    - **output_table** [string] The qualified name of the table to be created which will contain the model's predictions.

    - **schedule** [dict::]

        * **scheduled** [boolean] If the item is scheduled.

> > > > > > ∗ **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

> > > > > > ∗ **scheduled_hours** [list] Hours of the day it is scheduled on.

> > > > > > ∗ **scheduled_minutes** [list] Minutes of the day it is scheduled on.

> > > > > > ∗ **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

> > > > > > ∗ **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

> > > > > – **state** [string] The status of the prediction. One of: "succeeded", "failed", "queued", or "running,"or "idle", if no build has been attempted.

> > > > • **last_output_location** [string] The output JSON for the last build.
> > > > • **archived** [string] The archival status of the requested item(s).

get_builds(*id*, *build_id*)

> Check status of a build
> > **Parameters**
> > > **id** [integer] The ID of the model.
> > > **build_id** [integer] The ID of the build.
> > **Returns**
> > > [*civis.response.Response*](#)
> > > > • **id** [integer] The ID of the model build.
> > > > • **state** [string] The state of the model build.one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > > • **error** [string] The error, if any, returned by the build.
> > > > • **name** [string] The name of the model build.
> > > > • **created_at** [string] The time the model build was created.
> > > > • **description** [string] A description of the model build.
> > > > • **root_mean_squared_error** [number/float] A key metric for continuous models. Nil for other model types.
> > > > • **r_squared_error** [number/float] A key metric for continuous models. Nil for other model types.
> > > > • **roc_auc** [number/float] A key metric for binary, multinomial, and ordinal models. Nil for other model types.
> > > > • **transformation_metadata** [string] A string representing the full JSON output of the metadata for transformation of column names
> > > > • **output** [string] A string representing the JSON output for the specified build. Only present when smaller than 10KB in size.
> > > > • **output_location** [string] A URL representing the location of the full JSON output for the specified build.The URL link will be valid for 5 minutes.

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

> List
> > **Parameters**

> **model_name** [string, optional] If specified, will be used to filter the models returned. Substring matching is supported. (e.g., "modelName=model" will return both "model1" and "my model").
>
> **training_table_name** [string, optional] If specified, will be used to filter the models returned by the training dataset table name. Substring matching is supported. (e.g., "trainingTableName=table" will return both "table1" and "my_table").
>
> **dependent_variable** [string, optional] If specified, will be used to filter the models returned by the dependent variable column name. Substring matching is supported. (e.g., "dependentVariable=predictor" will return both "predictor" and "my predictor").
>
> **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'.
>
> **author** [string, optional] If specified, return items from any of these authors. It accepts a comma- separated list of user IDs.
>
> **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
>
> **archived** [string, optional] The archival status of the requested item(s).
>
> **limit** [integer, optional] Number of results to return. Defaults to its maximum of 50.
>
> **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
>
> **order** [string, optional] The field on which to order the result set. Defaults to updated_at. Must be one of: updated_at, name, created_at, last_run.updated_at.
>
> **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
>
> **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
>
> **Returns**
>
> *civis.response.PaginatedResponse*
>
> > - **id** [integer] The ID of the model.
> > - **table_name** [string] The qualified name of the table containing the training set from which to build the model.
> > - **database_id** [integer] The ID of the database holding the training set table used to build the model.
> > - **credential_id** [integer] The ID of the credential used to read the target table. Defaults to the user's default credential.
> > - **model_name** [string] The name of the model.
> > - **description** [string] A description of the model.
> > - **interaction_terms** [boolean] Whether to search for interaction terms.
> > - **box_cox_transformation** [boolean] Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.
> > - **model_type_id** [integer] The ID of the model's type.
> > - **primary_key** [string] The unique ID (primary key) of the training dataset.
> > - **dependent_variable** [string] The dependent variable of the training dataset.
> > - **dependent_variable_order** [list] The order of dependent variables, especially useful for Ordinal Modeling.
> > - **excluded_columns** [list] A list of columns which will be considered ineligible to be independent variables.
> > - **limiting_sql** [string] A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., "id > 105").
> > - **cross_validation_parameters** [dict] Cross validation parameter grid for tree methods, e.g. {"n_estimators": [100, 200, 500], "learn-

> > > ing_rate": [0.01, 0.1], "max_depth": [2, 3]}.

- **number_of_folds** [integer] Number of folds for cross validation. Default value is 5.
- **schedule** [dict::]

  - **scheduled** [boolean] If the item is scheduled.

  - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  - **scheduled_hours** [list] Hours of the day it is scheduled on.

  - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

  - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

  - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **parent_id** [integer] The ID of the parent job that will trigger this model.
- **time_zone** [string] The time zone of this model.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.
- **user** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **created_at** [string/date-time] The time the model was created.
- **updated_at** [string/date-time] The time the model was updated.
- **current_build_state** [string] The status of the current model build. One of "succeeded", "failed", "queued", or "running,"or "idle", if no build has been attempted.
- **current_build_exception** [string] Exception message, if applicable, of the current model build.
- **builds** [list::] A list of trained models available for making predictions. - id : integer

  > The ID of the model build.

  - **name** [string] The name of the model build.

  - **created_at** [string] The time the model build was created.

  - **description** [string] A description of the model build.

> – **root_mean_squared_error** [number/float] A key metric for
> continuous models. Nil for other model types.
>
> – **r_squared_error** [number/float] A key metric for continuous
> models. Nil for other model types.
>
> – **roc_auc** [number/float] A key metric for binary, multinomial,
> and ordinal models. Nil for other model types.

- **predictions** [list::] The tables upon which the model will be applied. - id
  : integer

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

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

- **last_output_location** [string] The output JSON for the last build.
- **archived** [string] The archival status of the requested item(s).

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

> List builds for the given model
>
> **Parameters**
>
> > **id** [integer] The ID of the model.
> >
> > **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed
> > is 100.
> >
> > **page_num** [integer, optional] Page number of the results to return. Defaults to the first
> > page, 1.
> >
> > **order** [string, optional] The field on which to order the result set. Defaults to id. Must
> > be one of: id.
> >
> > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc
> > (descending) defaulting to desc.
> >
> > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use
> > when more results than the maximum allowed by limit are needed. When True,
> > limit and page_num are ignored. Defaults to False.
>
> **Returns**
>
> > *civis.response.PaginatedResponse*
> >
> > > - **id** [integer] The ID of the model build.
> > > - **state** [string] The state of the model build.one of 'queued' 'running' 'suc-
> > >   ceeded' 'failed' or 'cancelled'.
> > > - **error** [string] The error, if any, returned by the build.
> > > - **name** [string] The name of the model build.
> > > - **created_at** [string] The time the model build was created.
> > > - **description** [string] A description of the model build.

- **root_mean_squared_error** [number/float] A key metric for continuous models. Nil for other model types.
- **r_squared_error** [number/float] A key metric for continuous models. Nil for other model types.
- **roc_auc** [number/float] A key metric for binary, multinomial, and ordinal models. Nil for other model types.
- **transformation_metadata** [string] A string representing the full JSON output of the metadata for transformation of column names
- **output** [string] A string representing the JSON output for the specified build. Only present when smaller than 10KB in size.
- **output_location** [string] A URL representing the location of the full JSON output for the specified build.The URL link will be valid for 5 minutes.

list_builds_logs(*id*, *build_id*, *, *last_id='DEFAULT'*, *limit='DEFAULT'*)

    Get the logs for a build

        Parameters

            id [integer] The ID of the model.

            build_id [integer] The ID of the build.

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

            limit [integer, optional] The maximum number of log messages to return. Default of 10000.

        Returns

        civis.response.Response

- **id** [integer] The ID of the log.
- **created_at** [string/date-time] The time the log was created.
- **message** [string] The log message.
- **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

list_dependencies(*id*, *, *user_id='DEFAULT'*)

    List dependent objects for this object

        Parameters

            id [integer] The ID of the resource that is shared.

            user_id [integer, optional] ID of target user

        Returns

        civis.response.Response

- **object_type** [string] Dependent object type
- **fco_type** [string] Human readable dependent object type
- **id** [integer] Dependent object ID
- **name** [string] Dependent object name, or nil if the requesting user cannot read this object
- **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
- **shareable** [boolean] Whether or not the requesting user can share this object.

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

    List the projects a Model belongs to

        Parameters

            id [integer] The ID of the Model.

            hidden [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.

> **Returns**
>> *civis.response.Response*
>>> - **id** [integer] The ID for this project.
>>> - **author** [dict::]
>>>> – **id** [integer] The ID of this user.
>>>>
>>>> – **name** [string] This user's name.
>>>>
>>>> – **username** [string] This user's username.
>>>>
>>>> – **initials** [string] This user's initials.
>>>>
>>>> – **online** [boolean] Whether this user is online.
>>> - **name** [string] The name of this project.
>>> - **description** [string] A description of the project.
>>> - **users** [list::] Users who can see the project. - id : integer
>>>> The ID of this user.
>>>>
>>>> – **name** [string] This user's name.
>>>>
>>>> – **username** [string] This user's username.
>>>>
>>>> – **initials** [string] This user's initials.
>>>>
>>>> – **online** [boolean] Whether this user is online.
>>> - auto_share : boolean
>>> - created_at : string/time
>>> - updated_at : string/time
>>> - **archived** [string] The archival status of the requested item(s).

list_schedules(*id*)

> Show the model build schedule
>> **Parameters**
>>> **id** [integer] The ID of the model associated with this schedule.
>>
>> **Returns**
>>> *civis.response.Response*
>>>> - **id** [integer] The ID of the model associated with this schedule.
>>>> - **schedule** [dict::]
>>>>> – **scheduled** [boolean] If the item is scheduled.
>>>>>
>>>>> – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
>>>>>
>>>>> – **scheduled_hours** [list] Hours of the day it is scheduled on.
>>>>>
>>>>> – **scheduled_minutes** [list] Minutes of the day it is scheduled on.
>>>>>
>>>>> – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
>>>>>
>>>>> – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

list_shares(*id*)

> List users and groups permissioned on this object
>> **Parameters**
>>> **id** [integer] The ID of the resource that is shared.

> **Returns**
> *civis.response.Response*
> - **readers** [dict::]
>   - **users** [list::]
>   * id : integer
>   * name : string
>   - **groups** [list::]
>   * id : integer
>   * name : string
> - **writers** [dict::]
>   - **users** [list::]
>   * id : integer
>   * name : string
>   - **groups** [list::]
>   * id : integer
>   * name : string
> - **owners** [dict::]
>   - **users** [list::]
>   * id : integer
>   * name : string
>   - **groups** [list::]
>   * id : integer
>   * name : string
> - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_types()

List all available model types

> **Returns**
> *civis.response.Response*
> - **id** [integer] The ID of the model type.
> - **algorithm** [string] The name of the algorithm used to train the model.
> - **dv_type** [string] The type of dependent variable predicted by the model.
> - **fint_allowed** [boolean] Whether this model type supports searching for interaction terms.

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**
> *civis.response.Response*
> - **id** [integer] The ID of the model.

- **table_name** [string] The qualified name of the table containing the training set from which to build the model.
- **database_id** [integer] The ID of the database holding the training set table used to build the model.
- **credential_id** [integer] The ID of the credential used to read the target table. Defaults to the user's default credential.
- **model_name** [string] The name of the model.
- **description** [string] A description of the model.
- **interaction_terms** [boolean] Whether to search for interaction terms.
- **box_cox_transformation** [boolean] Whether to transform data so that it assumes a normal distribution. Valid only with continuous models.
- **model_type_id** [integer] The ID of the model's type.
- **primary_key** [string] The unique ID (primary key) of the training dataset.
- **dependent_variable** [string] The dependent variable of the training dataset.
- **dependent_variable_order** [list] The order of dependent variables, especially useful for Ordinal Modeling.
- **excluded_columns** [list] A list of columns which will be considered ineligible to be independent variables.
- **limiting_sql** [string] A custom SQL WHERE clause used to filter the rows used to build the model. (e.g., "id > 105").
- **active_build_id** [integer] The ID of the current active build, the build used to score predictions.
- **cross_validation_parameters** [dict] Cross validation parameter grid for tree methods, e.g. {"n_estimators": [100, 200, 500], "learning_rate": [0.01, 0.1], "max_depth": [2, 3]}.
- **number_of_folds** [integer] Number of folds for cross validation. Default value is 5.
- **notifications** [dict::]

    – **urls** [list] URLs to receive a POST request at job completion

    – **success_email_subject** [string] Custom subject line for success e-mail.

    – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    – **success_on** [boolean] If success email notifications are on.

    – **failure_on** [boolean] If failure email notifications are on.
- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

– **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

– **scheduled_hours** [list] Hours of the day it is scheduled on.

– **scheduled_minutes** [list] Minutes of the day it is scheduled on.

– **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

– **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **parent_id** [integer] The ID of the parent job that will trigger this model.
- **running_as** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.

- **time_zone** [string] The time zone of this model.
- **last_run** [dict::]

  – id : integer

  – state : string

  – **created_at** [string/time] The time that the run was queued.

  – **started_at** [string/time] The time that the run started.

  – **finished_at** [string/time] The time that the run completed.

  – **error** [string] The error message for this run, if present.

- **hidden** [boolean] The hidden status of the item.
- **user** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.

- **created_at** [string/date-time] The time the model was created.
- **updated_at** [string/date-time] The time the model was updated.
- **current_build_state** [string] The status of the current model build. One of "succeeded", "failed", "queued", or "running,"or "idle", if no build has been attempted.
- **current_build_exception** [string] Exception message, if applicable, of the current model build.
- **builds** [list::] A list of trained models available for making predictions. - id : integer

  The ID of the model build.

- **name** [string] The name of the model build.

- **created_at** [string] The time the model build was created.

- **description** [string] A description of the model build.

- **root_mean_squared_error** [number/float] A key metric for continuous models. Nil for other model types.

- **r_squared_error** [number/float] A key metric for continuous models. Nil for other model types.

- **roc_auc** [number/float] A key metric for binary, multinomial, and ordinal models. Nil for other model types.

- **predictions** [list::] The tables upon which the model will be applied. - id : integer

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

    - **table_name** [string] The qualified name of the table on which to apply the predictive model.

    - **primary_key** [list] The primary key or composite keys of the table being predicted.

    - **limiting_sql** [string] A SQL WHERE clause used to scope the rows to be predicted.

    - **output_table** [string] The qualified name of the table to be created which will contain the model's predictions.

    - **schedule** [dict::]

        * **scheduled** [boolean] If the item is scheduled.

        * **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

        * **scheduled_hours** [list] Hours of the day it is scheduled on.

        * **scheduled_minutes** [list] Minutes of the day it is scheduled on.

        * **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

        * **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

    - **state** [string] The status of the prediction. One of: "succeeded", "failed", "queued", or "running,"or "idle", if no build has been attempted.

- **last_output_location** [string] The output JSON for the last build.
- **archived** [string] The archival status of the requested item(s).

put_projects(*id*, *project_id*)
> Add a Model to a project
> > **Parameters**

> **id** [integer] The ID of the Model.
>
> **project_id** [integer] The ID of the project.
>
> **Returns**
>
> > **None** Response code 204: success

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

> Set the permissions groups has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **group_ids** [list] An array of one or more group IDs.
> > >
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > >
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > >
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > [*civis.response.Response*](#)
> >
> > > - **readers** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > > - **writers** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > > - **owners** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > >
> > > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object
>
> > **Parameters**

id [integer] The ID of the resource that is shared.

user_ids [list] An array of one or more user IDs.

permission_level [string] Options are: "read", "write", or "manage".

share_email_body [string, optional] Custom body text for e-mail sent on a share.

send_shared_email [boolean, optional] Send email to the recipients of a share.

Returns

civis.response.Response

- **readers** [dict::]

  – **users** [list::]

    ∗ id : integer

    ∗ name : string

  – **groups** [list::]

    ∗ id : integer

    ∗ name : string

- **writers** [dict::]

  – **users** [list::]

    ∗ id : integer

    ∗ name : string

  – **groups** [list::]

    ∗ id : integer

    ∗ name : string

- **owners** [dict::]

  – **users** [list::]

    ∗ id : integer

    ∗ name : string

  – **groups** [list::]

    ∗ id : integer

    ∗ name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_transfer(*id*, *user_id*, *include_dependencies*, *, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

Transfer ownership of this object to another user

Parameters

id [integer] The ID of the resource that is shared.

user_id [integer] ID of target user

include_dependencies [boolean] Whether or not to give manage permissions on all dependencies

email_body [string, optional] Custom body text for e-mail sent on transfer.

send_email [boolean, optional] Send email to the target user of the transfer?

Returns

civis.response.Response

- **dependencies** [list::] Dependent objects for this object - object_type : string

    Dependent object type

    – **fco_type** [string] Human readable dependent object type

    – **id** [integer] Dependent object ID

    – **name** [string] Dependent object name, or nil if the requesting user cannot read this object

    – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user

    – **shared** [boolean] Whether dependent object was successfully shared with target user

Notebooks

class **Notebooks**(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.notebooks.list(...)
```

Methods

delete_deployments(notebook_id, deployment_id)	Delete a Notebook deployment
delete_projects(id, project_id)	Remove a Notebook 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 a Notebook
get_deployments(notebook_id, deployment_id)	Get details about a Notebook deployment
get_git_commits(id, commit_hash)	Get file contents at git ref
list(*[, hidden, archived, author, status, ...])	List Notebooks
list_dependencies(id, *[, user_id])	List dependent objects for this object
list_deployments(notebook_id, *[, ...])	List deployments for a Notebook
list_deployments_logs(id, deployment_id, *)	Get the logs for a Notebook deployment
list_git(id)	Get the git metadata attached to an item
list_git_commits(id)	Get the git commits for an item on the current branch
list_projects(id, *[, hidden])	List the projects a Notebook belongs to
list_shares(id)	List users and groups permissioned on this object
list_update_links(id)	Get URLs to update notebook
patch(id, *[, name, language, description, ...])	Update some attributes of this Notebook
patch_git(id, *[, git_ref, git_branch, ...])	Update an attached git file

continues on next page

Table 43 – continued from previous page

post(*[, name, language, description, ...])	Create a Notebook
post_clone(id)	Clone this Notebook
post_deployments(notebook_id, *[, deployment_id])	Deploy a Notebook
post_git_checkout(id)	Checkout content that the existing git_ref points to and save to the object
post_git_checkout_latest(id)	Checkout latest commit on the current branch of a script or workflow
post_git_commits(id, content, message, file_hash)	Commit and push a new version of the file
put(id, *[, name, language, description, ...])	Replace all attributes of this Notebook
put_archive(id, status)	Update the archive status of this object
put_git(id, *[, git_ref, git_branch, ...])	Attach an item to a file in a git repo
put_projects(id, project_id)	Add a Notebook to a project
put_shares_groups(id, group_ids, ...[, ...])	Set the permissions groups has on this object
put_shares_users(id, user_ids, ...[, ...])	Set the permissions users have on this object
put_transfer(id, user_id, ...[, email_body, ...])	Transfer ownership of this object to another user

—

delete_deployments(*notebook_id*, *deployment_id*)

> Delete a Notebook deployment
>
> > **Parameters**
> >
> > > **notebook_id** [integer] The ID of the owning Notebook
> > >
> > > **deployment_id** [integer] The ID for this deployment
> >
> > **Returns**
> >
> > > **None** Response code 204: success

delete_projects(*id*, *project_id*)

> Remove a Notebook from a project
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the Notebook.
> > >
> > > **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] The ID of the resource that is shared.
> > >
> > > **group_id** [integer] The 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] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] The ID of the user.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

get(*id*)

> Get a Notebook

> **Parameters**
>> **id** [integer]
>
> **Returns**
>> *civis.response.Response*
>
>> - **id** [integer] The ID for this notebook.
>> - **name** [string] The name of this notebook.
>> - **language** [string] The kernel language of this notebook.
>> - **description** [string] The description of this notebook.
>> - **notebook_url** [string] Time-limited URL to get the .ipynb file for this notebook.
>> - **notebook_preview_url** [string] Time-limited URL to get the .htm preview file for this notebook.
>> - **requirements_url** [string] Time-limited URL to get the requirements.txt file for this notebook.
>> - **file_id** [string] The file ID for the S3 file containing the .ipynb file.
>> - **requirements_file_id** [string] The file ID for the S3 file containing the requirements.txt file.
>> - **user** [dict::]
>>
>>> - **id** [integer] The ID of this user.
>>>
>>> - **name** [string] This user's name.
>>>
>>> - **username** [string] This user's username.
>>>
>>> - **initials** [string] This user's initials.
>>>
>>> - **online** [boolean] Whether this user is online.
>>
>> - **docker_image_name** [string] The name of the docker image to pull from DockerHub.
>> - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
>> - **instance_type** [string] The EC2 instance type to deploy to.
>> - **memory** [integer] The amount of memory allocated to the notebook.
>> - **cpu** [integer] The amount of cpu allocated to the the notebook.
>> - created_at : string/time
>> - updated_at : string/time
>> - **most_recent_deployment** [dict::]
>>
>>> - **deployment_id** [integer] The ID for this deployment.
>>>
>>> - **user_id** [integer] The ID of the owner.
>>>
>>> - **host** [string] Domain of the deployment.
>>>
>>> - **name** [string] Name of the deployment.
>>>
>>> - **docker_image_name** [string] The name of the docker image to pull from DockerHub.
>>>
>>> - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
>>>
>>> - **display_url** [string] A signed URL for viewing the deployed item.
>>>
>>> - **instance_type** [string] The EC2 instance type requested for the deployment.
>>>
>>> - **memory** [integer] The memory allocated to the deployment, in MB.

> - **cpu** [integer] The cpu allocated to the deployment, in millicores.
>
> - **state** [string] The state of the deployment.
>
> - **state_message** [string] A detailed description of the state.
>
> - **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.
>
> - **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.
>
> - created_at : string/time
>
> - updated_at : string/time
>
> - **notebook_id** [integer] The ID of owning Notebook
- **credentials** [list] A list of credential IDs to pass to the notebook.
- **environment_variables** [dict] Environment variables to be passed into the Notebook.
- **idle_timeout** [integer] How long the notebook will stay alive without any kernel activity.
- **partition_label** [string] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.
- **git_repo_id** [integer] The ID of the git repository.
- **git_repo_url** [string] The url of the git repository
- **git_ref** [string] The git reference if git repo is specified
- **git_path** [string] The path to the .ipynb file in the git repo that will be started up on notebook launch
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.

get_deployments(*notebook_id*, *deployment_id*)
> Get details about a Notebook deployment
>
> > **Parameters**
> >
> > > **notebook_id** [integer] The ID of the owning Notebook
> > >
> > > **deployment_id** [integer] The ID for this deployment
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > - **deployment_id** [integer] The ID for this deployment.
> > > > - **user_id** [integer] The ID of the owner.
> > > > - **host** [string] Domain of the deployment.
> > > > - **name** [string] Name of the deployment.
> > > > - **docker_image_name** [string] The name of the docker image to pull from DockerHub.
> > > > - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
> > > > - **display_url** [string] A signed URL for viewing the deployed item.
> > > > - **instance_type** [string] The EC2 instance type requested for the deployment.
> > > > - **memory** [integer] The memory allocated to the deployment, in MB.
> > > > - **cpu** [integer] The cpu allocated to the deployment, in millicores.

> - **state** [string] The state of the deployment.
> - **state_message** [string] A detailed description of the state.
> - **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.
> - **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.
> - created_at : string/time
> - updated_at : string/time
> - **notebook_id** [integer] The ID of owning Notebook

get_git_commits(*id*, *commit_hash*)

> Get file contents at git ref
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the file.
> > >
> > > **commit_hash** [string] The SHA (full or shortened) of the desired git commit.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **content** [string] The file's contents.
> > > > - **type** [string] The file's type.
> > > > - **size** [integer] The file's size.
> > > > - **file_hash** [string] The SHA of the file.

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

> List Notebooks
>
> > **Parameters**
> >
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> > >
> > > **archived** [string, optional] The archival status of the requested item(s).
> > >
> > > **author** [string, optional] If specified, return items from any of these authors. It accepts a comma- separated list of user IDs.
> > >
> > > **status** [string, optional] If specified, returns notebooks with one of these statuses. It accepts a comma-separated list, possible values are 'running', 'pending', 'idle'.
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 50.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to updated_at. Must be one of: updated_at, name, created_at.
> > >
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > >
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> >
> > **Returns**
> >
> > > *civis.response.PaginatedResponse*
> > >
> > > > - **id** [integer] The ID for this notebook.
> > > > - **name** [string] The name of this notebook.
> > > > - **language** [string] The kernel language of this notebook.
> > > > - **description** [string] The description of this notebook.
> > > > - **user** [dict::]
> > > >
> > > > > - **id** [integer] The ID of this user.
> > > > >
> > > > > - **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- created_at : string/time
- updated_at : string/time
- **most_recent_deployment** [dict::]

  - **deployment_id** [integer] The ID for this deployment.

  - **user_id** [integer] The ID of the owner.

  - **host** [string] Domain of the deployment.

  - **name** [string] Name of the deployment.

  - **docker_image_name** [string] The name of the docker image to pull from DockerHub.

  - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).

  - **instance_type** [string] The EC2 instance type requested for the deployment.

  - **memory** [integer] The memory allocated to the deployment, in MB.

  - **cpu** [integer] The cpu allocated to the deployment, in millicores.

  - **state** [string] The state of the deployment.

  - **state_message** [string] A detailed description of the state.

  - **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.

  - **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.

  - created_at : string/time

  - updated_at : string/time

  - **notebook_id** [integer] The ID of owning Notebook
- **archived** [string] The archival status of the requested item(s).

list_dependencies(*id*, *, *user_id='DEFAULT'*)

  List dependent objects for this object

  Parameters

  id [integer] The ID of the resource that is shared.

  user_id [integer, optional] ID of target user

  Returns

  civis.response.Response

  - **object_type** [string] Dependent object type
  - **fco_type** [string] Human readable dependent object type
  - **id** [integer] Dependent object ID
  - **name** [string] Dependent object name, or nil if the requesting user cannot read this object

---

- **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
- **shareable** [boolean] Whether or not the requesting user can share this object.

list_deployments(*notebook_id*, *, *deployment_id='DEFAULT'*, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)

List deployments for a Notebook

> **Parameters**
>
> > **notebook_id** [integer] The ID of the owning Notebook
> >
> > **deployment_id** [integer, optional] The ID for this deployment
> >
> > **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 50.
> >
> > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> >
> > **order** [string, optional] The field on which to order the result set. Defaults to created_at. Must be one of: created_at.
> >
> > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> >
> > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
>
> **Returns**
>
> > *civis.response.PaginatedResponse*
> >
> > > - **deployment_id** [integer] The ID for this deployment.
> > > - **user_id** [integer] The ID of the owner.
> > > - **host** [string] Domain of the deployment.
> > > - **name** [string] Name of the deployment.
> > > - **docker_image_name** [string] The name of the docker image to pull from DockerHub.
> > > - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
> > > - **instance_type** [string] The EC2 instance type requested for the deployment.
> > > - **memory** [integer] The memory allocated to the deployment, in MB.
> > > - **cpu** [integer] The cpu allocated to the deployment, in millicores.
> > > - **state** [string] The state of the deployment.
> > > - **state_message** [string] A detailed description of the state.
> > > - **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.
> > > - **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.
> > > - created_at : string/time
> > > - updated_at : string/time
> > > - **notebook_id** [integer] The ID of owning Notebook

list_deployments_logs(*id*, *deployment_id*, *, *start_at='DEFAULT'*, *end_at='DEFAULT'*, *limit='DEFAULT'*)

Get the logs for a Notebook deployment

> **Parameters**
>
> > **id** [integer] The ID of the owning Notebook.
> >
> > **deployment_id** [integer] The ID for this deployment.
> >
> > **start_at** [string, optional] Log entries with a lower timestamp will be omitted.
> >
> > **end_at** [string, optional] Log entries with a higher timestamp will be omitted.
> >
> > **limit** [integer, optional] The maximum number of log messages to return. Default of

---

> 10000.

> **Returns**
>> *civis.response.Response*
>>> • **message** [string] The log message.
>>> • **stream** [string] The stream of the log. One of "stdout", "stderr".
>>> • **created_at** [string/date-time] The time the log was created.
>>> • **source** [string] The source of the log. One of "system", "user".

list_git(*id*)

> Get the git metadata attached to an item

>> **Parameters**
>>> **id** [integer] The ID of the file.

>> **Returns**
>>> *civis.response.Response*
>>>> • **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
>>>> • **git_branch** [string] The git branch that the file is on.
>>>> • **git_path** [string] The path of the file in the repository.
>>>> • **git_repo** [dict::]
>>>>
>>>>> – **id** [integer] The ID for this git repository.
>>>>>
>>>>> – **repo_url** [string] The URL for this git repository.
>>>>>
>>>>> – created_at : string/time
>>>>>
>>>>> – updated_at : string/time
>>>> • **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
>>>> • **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

list_git_commits(*id*)

> Get the git commits for an item on the current branch

>> **Parameters**
>>> **id** [integer] The ID of the file.

>> **Returns**
>>> *civis.response.Response*
>>>> • **commit_hash** [string] The SHA of the commit.
>>>> • **author_name** [string] The name of the commit's author.
>>>> • **date** [string/time] The commit's timestamp.
>>>> • **message** [string] The commit message.

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

> List the projects a Notebook belongs to

>> **Parameters**
>>> **id** [integer] The ID of the Notebook.
>>> **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.

>> **Returns**
>>> *civis.response.Response*
>>>> • **id** [integer] The ID for this project.
>>>> • **author** [dict::]
>>>>
>>>>> – **id** [integer] The ID of this user.
>>>>>
>>>>> – **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **name** [string] The name of this project.
- **description** [string] A description of the project.
- **users** [list::] Users who can see the project. - id : integer

    The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- auto_share : boolean
- created_at : string/time
- updated_at : string/time
- **archived** [string] The archival status of the requested item(s).

list_shares(*id*)

List users and groups permissioned on this object

    Parameters

        id [integer] The ID of the resource that is shared.

    Returns

        civis.response.Response

- **readers** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **writers** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **owners** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string
> > > >
> > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_update_links(*id*)

> Get URLs to update notebook
>
> > **Parameters**
> >
> > > **id** [integer]
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **update_url** [string] Time-limited URL to PUT new contents of the .ipynb file for this notebook.
> > > > • **update_preview_url** [string] Time-limited URL to PUT new contents of the .htm preview file for this notebook.

patch(*id*, *, *name='DEFAULT'*, *language='DEFAULT'*, *description='DEFAULT'*, *file_id='DEFAULT'*, *requirements_file_id='DEFAULT'*, *requirements='DEFAULT'*, *docker_image_name='DEFAULT'*, *docker_image_tag='DEFAULT'*, *instance_type='DEFAULT'*, *memory='DEFAULT'*, *cpu='DEFAULT'*, *credentials='DEFAULT'*, *environment_variables='DEFAULT'*, *idle_timeout='DEFAULT'*, *partition_label='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref='DEFAULT'*, *git_path='DEFAULT'*)

> Update some attributes of this Notebook
>
> > **Parameters**
> >
> > > **id** [integer] The ID for this notebook.
> > >
> > > **name** [string, optional] The name of this notebook.
> > >
> > > **language** [string, optional] The kernel language of this notebook.
> > >
> > > **description** [string, optional] The description of this notebook.
> > >
> > > **file_id** [string, optional] The file ID for the S3 file containing the .ipynb file.
> > >
> > > **requirements_file_id** [string, optional] The file ID for the S3 file containing the requirements.txt file.
> > >
> > > **requirements** [string, optional] The requirements txt file.
> > >
> > > **docker_image_name** [string, optional] The name of the docker image to pull from DockerHub.
> > >
> > > **docker_image_tag** [string, optional] The tag of the docker image to pull from DockerHub (default: latest).
> > >
> > > **instance_type** [string, optional] The EC2 instance type to deploy to.
> > >
> > > **memory** [integer, optional] The amount of memory allocated to the notebook.
> > >
> > > **cpu** [integer, optional] The amount of cpu allocated to the the notebook.
> > >
> > > **credentials** [list, optional] A list of credential IDs to pass to the notebook.
> > >
> > > **environment_variables** [dict, optional] Environment variables to be passed into the Notebook.
> > >
> > > **idle_timeout** [integer, optional] How long the notebook will stay alive without any kernel activity.
> > >
> > > **partition_label** [string, optional] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.
> > >
> > > **git_repo_url** [string, optional] The url of the git repository
> > >
> > > **git_ref** [string, optional] The git reference if git repo is specified
> > >
> > > **git_path** [string, optional] The path to the .ipynb file in the git repo that will be started up on notebook launch
> >
> > **Returns**
> >
> > > *civis.response.Response*

- **id** [integer] The ID for this notebook.
- **name** [string] The name of this notebook.
- **language** [string] The kernel language of this notebook.
- **description** [string] The description of this notebook.
- **notebook_url** [string] Time-limited URL to get the .ipynb file for this notebook.
- **notebook_preview_url** [string] Time-limited URL to get the .htm preview file for this notebook.
- **requirements_url** [string] Time-limited URL to get the requirements.txt file for this notebook.
- **file_id** [string] The file ID for the S3 file containing the .ipynb file.
- **requirements_file_id** [string] The file ID for the S3 file containing the requirements.txt file.
- **user** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **docker_image_name** [string] The name of the docker image to pull from DockerHub.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
- **instance_type** [string] The EC2 instance type to deploy to.
- **memory** [integer] The amount of memory allocated to the notebook.
- **cpu** [integer] The amount of cpu allocated to the the notebook.
- created_at : string/time
- updated_at : string/time
- **most_recent_deployment** [dict::]

    - **deployment_id** [integer] The ID for this deployment.

    - **user_id** [integer] The ID of the owner.

    - **host** [string] Domain of the deployment.

    - **name** [string] Name of the deployment.

    - **docker_image_name** [string] The name of the docker image to pull from DockerHub.

    - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).

    - **display_url** [string] A signed URL for viewing the deployed item.

    - **instance_type** [string] The EC2 instance type requested for the deployment.

    - **memory** [integer] The memory allocated to the deployment, in MB.

    - **cpu** [integer] The cpu allocated to the deployment, in millicores.

    - **state** [string] The state of the deployment.

- **state_message** [string] A detailed description of the state.

- **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.

- **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.

- created_at : string/time

- updated_at : string/time

- **notebook_id** [integer] The ID of owning Notebook

- **credentials** [list] A list of credential IDs to pass to the notebook.
- **environment_variables** [dict] Environment variables to be passed into the Notebook.
- **idle_timeout** [integer] How long the notebook will stay alive without any kernel activity.
- **partition_label** [string] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.
- **git_repo_id** [integer] The ID of the git repository.
- **git_repo_url** [string] The url of the git repository
- **git_ref** [string] The git reference if git repo is specified
- **git_path** [string] The path to the .ipynb file in the git repo that will be started up on notebook launch
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.

patch_git(*id*, ***, *git_ref='DEFAULT'*, *git_branch='DEFAULT'*, *git_path='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref_type='DEFAULT'*, *pull_from_git='DEFAULT'*)

Update an attached git file

Parameters

id [integer] The ID of the file.

git_ref [string, optional] A git reference specifying an unambiguous version of the file. Can be a branch name, or the full or shortened SHA of a commit.

git_branch [string, optional] The git branch that the file is on.

git_path [string, optional] The path of the file in the repository.

git_repo_url [string, optional] The URL of the git repository.

git_ref_type [string, optional] Specifies if the file is versioned by branch or tag.

pull_from_git [boolean, optional] Automatically pull latest commit from git. Only works for scripts.

Returns

civis.response.Response

- **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
- **git_branch** [string] The git branch that the file is on.
- **git_path** [string] The path of the file in the repository.
- **git_repo** [dict::]

  - **id** [integer] The ID for this git repository.

  - **repo_url** [string] The URL for this git repository.

> – created_at : string/time
>
> – updated_at : string/time

- **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
- **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

post(*, *name='DEFAULT'*, *language='DEFAULT'*, *description='DEFAULT'*, *file_id='DEFAULT'*, *requirements_file_id='DEFAULT'*, *requirements='DEFAULT'*, *docker_image_name='DEFAULT'*, *docker_image_tag='DEFAULT'*, *instance_type='DEFAULT'*, *memory='DEFAULT'*, *cpu='DEFAULT'*, *credentials='DEFAULT'*, *environment_variables='DEFAULT'*, *idle_timeout='DEFAULT'*, *partition_label='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref='DEFAULT'*, *git_path='DEFAULT'*, *hidden='DEFAULT'*)

Create a Notebook

> **Parameters**
>
>> **name** [string, optional] The name of this notebook.
>>
>> **language** [string, optional] The kernel language of this notebook.
>>
>> **description** [string, optional] The description of this notebook.
>>
>> **file_id** [string, optional] The file ID for the S3 file containing the .ipynb file.
>>
>> **requirements_file_id** [string, optional] The file ID for the S3 file containing the requirements.txt file.
>>
>> **requirements** [string, optional] The requirements txt file.
>>
>> **docker_image_name** [string, optional] The name of the docker image to pull from DockerHub.
>>
>> **docker_image_tag** [string, optional] The tag of the docker image to pull from DockerHub (default: latest).
>>
>> **instance_type** [string, optional] The EC2 instance type to deploy to.
>>
>> **memory** [integer, optional] The amount of memory allocated to the notebook.
>>
>> **cpu** [integer, optional] The amount of cpu allocated to the the notebook.
>>
>> **credentials** [list, optional] A list of credential IDs to pass to the notebook.
>>
>> **environment_variables** [dict, optional] Environment variables to be passed into the Notebook.
>>
>> **idle_timeout** [integer, optional] How long the notebook will stay alive without any kernel activity.
>>
>> **partition_label** [string, optional] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.
>>
>> **git_repo_url** [string, optional] The url of the git repository
>>
>> **git_ref** [string, optional] The git reference if git repo is specified
>>
>> **git_path** [string, optional] The path to the .ipynb file in the git repo that will be started up on notebook launch
>>
>> **hidden** [boolean, optional] The hidden status of the item.
>
> **Returns**
>
>> *civis.response.Response*
>>
>>> - **id** [integer] The ID for this notebook.
>>> - **name** [string] The name of this notebook.
>>> - **language** [string] The kernel language of this notebook.
>>> - **description** [string] The description of this notebook.
>>> - **notebook_url** [string] Time-limited URL to get the .ipynb file for this notebook.
>>> - **notebook_preview_url** [string] Time-limited URL to get the .htm preview file for this notebook.
>>> - **requirements_url** [string] Time-limited URL to get the requirements.txt file for this notebook.

- **file_id** [string] The file ID for the S3 file containing the .ipynb file.
- **requirements_file_id** [string] The file ID for the S3 file containing the requirements.txt file.
- **user** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- **docker_image_name** [string] The name of the docker image to pull from DockerHub.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
- **instance_type** [string] The EC2 instance type to deploy to.
- **memory** [integer] The amount of memory allocated to the notebook.
- **cpu** [integer] The amount of cpu allocated to the the notebook.
- created_at : string/time
- updated_at : string/time
- **most_recent_deployment** [dict::]

    – **deployment_id** [integer] The ID for this deployment.

    – **user_id** [integer] The ID of the owner.

    – **host** [string] Domain of the deployment.

    – **name** [string] Name of the deployment.

    – **docker_image_name** [string] The name of the docker image to pull from DockerHub.

    – **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).

    – **display_url** [string] A signed URL for viewing the deployed item.

    – **instance_type** [string] The EC2 instance type requested for the deployment.

    – **memory** [integer] The memory allocated to the deployment, in MB.

    – **cpu** [integer] The cpu allocated to the deployment, in millicores.

    – **state** [string] The state of the deployment.

    – **state_message** [string] A detailed description of the state.

    – **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.

    – **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.

    – created_at : string/time

- updated_at : string/time

- **notebook_id** [integer] The ID of owning Notebook
- **credentials** [list] A list of credential IDs to pass to the notebook.
- **environment_variables** [dict] Environment variables to be passed into the Notebook.
- **idle_timeout** [integer] How long the notebook will stay alive without any kernel activity.
- **partition_label** [string] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.
- **git_repo_id** [integer] The ID of the git repository.
- **git_repo_url** [string] The url of the git repository
- **git_ref** [string] The git reference if git repo is specified
- **git_path** [string] The path to the .ipynb file in the git repo that will be started up on notebook launch
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.

post_clone(*id*)

 Clone this Notebook

  Parameters

   id [integer]

  Returns

   civis.response.Response

- **id** [integer] The ID for this notebook.
- **name** [string] The name of this notebook.
- **language** [string] The kernel language of this notebook.
- **description** [string] The description of this notebook.
- **notebook_url** [string] Time-limited URL to get the .ipynb file for this notebook.
- **notebook_preview_url** [string] Time-limited URL to get the .htm preview file for this notebook.
- **requirements_url** [string] Time-limited URL to get the requirements.txt file for this notebook.
- **file_id** [string] The file ID for the S3 file containing the .ipynb file.
- **requirements_file_id** [string] The file ID for the S3 file containing the requirements.txt file.
- **user** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **docker_image_name** [string] The name of the docker image to pull from DockerHub.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
- **instance_type** [string] The EC2 instance type to deploy to.
- **memory** [integer] The amount of memory allocated to the notebook.

- **cpu** [integer] The amount of cpu allocated to the the notebook.
- created_at : string/time
- updated_at : string/time
- **most_recent_deployment** [dict::]

    – **deployment_id** [integer] The ID for this deployment.

    – **user_id** [integer] The ID of the owner.

    – **host** [string] Domain of the deployment.

    – **name** [string] Name of the deployment.

    – **docker_image_name** [string] The name of the docker image to pull from DockerHub.

    – **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).

    – **display_url** [string] A signed URL for viewing the deployed item.

    – **instance_type** [string] The EC2 instance type requested for the deployment.

    – **memory** [integer] The memory allocated to the deployment, in MB.

    – **cpu** [integer] The cpu allocated to the deployment, in millicores.

    – **state** [string] The state of the deployment.

    – **state_message** [string] A detailed description of the state.

    – **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.

    – **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.

    – created_at : string/time

    – updated_at : string/time

    – **notebook_id** [integer] The ID of owning Notebook
- **credentials** [list] A list of credential IDs to pass to the notebook.
- **environment_variables** [dict] Environment variables to be passed into the Notebook.
- **idle_timeout** [integer] How long the notebook will stay alive without any kernel activity.
- **partition_label** [string] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.
- **git_repo_id** [integer] The ID of the git repository.
- **git_repo_url** [string] The url of the git repository
- **git_ref** [string] The git reference if git repo is specified
- **git_path** [string] The path to the .ipynb file in the git repo that will be started up on notebook launch

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.

post_deployments(*notebook_id*, ***, *deployment_id='DEFAULT'*)

Deploy a Notebook

> **Parameters**
>> **notebook_id** [integer] The ID of the owning Notebook
>>
>> **deployment_id** [integer, optional] The ID for this deployment
>
> **Returns**
>> *civis.response.Response*
>>> - **deployment_id** [integer] The ID for this deployment.
>>> - **user_id** [integer] The ID of the owner.
>>> - **host** [string] Domain of the deployment.
>>> - **name** [string] Name of the deployment.
>>> - **docker_image_name** [string] The name of the docker image to pull from DockerHub.
>>> - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
>>> - **display_url** [string] A signed URL for viewing the deployed item.
>>> - **instance_type** [string] The EC2 instance type requested for the deployment.
>>> - **memory** [integer] The memory allocated to the deployment, in MB.
>>> - **cpu** [integer] The cpu allocated to the deployment, in millicores.
>>> - **state** [string] The state of the deployment.
>>> - **state_message** [string] A detailed description of the state.
>>> - **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.
>>> - **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.
>>> - created_at : string/time
>>> - updated_at : string/time
>>> - **notebook_id** [integer] The ID of owning Notebook

post_git_checkout(*id*)

Checkout content that the existing git_ref points to and save to the object

> **Parameters**
>> **id** [integer] The ID of the file.
>
> **Returns**
>> *civis.response.Response*
>>> - **content** [string] The file's contents.
>>> - **type** [string] The file's type.
>>> - **size** [integer] The file's size.
>>> - **file_hash** [string] The SHA of the file.

post_git_checkout_latest(*id*)

Checkout latest commit on the current branch of a script or workflow

> **Parameters**
>> **id** [integer] The ID of the file.
>
> **Returns**
>> *civis.response.Response*
>>> - **content** [string] The file's contents.
>>> - **type** [string] The file's type.
>>> - **size** [integer] The file's size.

> • **file_hash** [string] The SHA of the file.

post_git_commits(*id*, *content*, *message*, *file_hash*)

    Commit and push a new version of the file

        Parameters

            id [integer] The ID of the file.

            content [string] The contents to commit to the file.

            message [string] A commit message describing the changes being made.

            file_hash [string] The full SHA of the file being replaced.

        Returns

            civis.response.Response

> • **content** [string] The file's contents.
> • **type** [string] The file's type.
> • **size** [integer] The file's size.
> • **file_hash** [string] The SHA of the file.

put(*id*, *, *name='DEFAULT'*, *language='DEFAULT'*, *description='DEFAULT'*, *file_id='DEFAULT'*, *requirements_file_id='DEFAULT'*, *requirements='DEFAULT'*, *docker_image_name='DEFAULT'*, *docker_image_tag='DEFAULT'*, *instance_type='DEFAULT'*, *memory='DEFAULT'*, *cpu='DEFAULT'*, *credentials='DEFAULT'*, *environment_variables='DEFAULT'*, *idle_timeout='DEFAULT'*, *partition_label='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref='DEFAULT'*, *git_path='DEFAULT'*)

    Replace all attributes of this Notebook

        Parameters

            id [integer] The ID for this notebook.

            name [string, optional] The name of this notebook.

            language [string, optional] The kernel language of this notebook.

            description [string, optional] The description of this notebook.

            file_id [string, optional] The file ID for the S3 file containing the .ipynb file.

            requirements_file_id [string, optional] The file ID for the S3 file containing the requirements.txt file.

            requirements [string, optional] The requirements txt file.

            docker_image_name [string, optional] The name of the docker image to pull from DockerHub.

            docker_image_tag [string, optional] The tag of the docker image to pull from DockerHub (default: latest).

            instance_type [string, optional] The EC2 instance type to deploy to.

            memory [integer, optional] The amount of memory allocated to the notebook.

            cpu [integer, optional] The amount of cpu allocated to the the notebook.

            credentials [list, optional] A list of credential IDs to pass to the notebook.

            environment_variables [dict, optional] Environment variables to be passed into the Notebook.

            idle_timeout [integer, optional] How long the notebook will stay alive without any kernel activity.

            partition_label [string, optional] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.

            git_repo_url [string, optional] The url of the git repository

            git_ref [string, optional] The git reference if git repo is specified

            git_path [string, optional] The path to the .ipynb file in the git repo that will be started up on notebook launch

        Returns

            civis.response.Response

> • **id** [integer] The ID for this notebook.
> • **name** [string] The name of this notebook.
> • **language** [string] The kernel language of this notebook.

- **description** [string] The description of this notebook.
- **notebook_url** [string] Time-limited URL to get the .ipynb file for this notebook.
- **notebook_preview_url** [string] Time-limited URL to get the .htm preview file for this notebook.
- **requirements_url** [string] Time-limited URL to get the requirements.txt file for this notebook.
- **file_id** [string] The file ID for the S3 file containing the .ipynb file.
- **requirements_file_id** [string] The file ID for the S3 file containing the requirements.txt file.
- **user** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **docker_image_name** [string] The name of the docker image to pull from DockerHub.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
- **instance_type** [string] The EC2 instance type to deploy to.
- **memory** [integer] The amount of memory allocated to the notebook.
- **cpu** [integer] The amount of cpu allocated to the the notebook.
- created_at : string/time
- updated_at : string/time
- **most_recent_deployment** [dict::]

    - **deployment_id** [integer] The ID for this deployment.

    - **user_id** [integer] The ID of the owner.

    - **host** [string] Domain of the deployment.

    - **name** [string] Name of the deployment.

    - **docker_image_name** [string] The name of the docker image to pull from DockerHub.

    - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).

    - **display_url** [string] A signed URL for viewing the deployed item.

    - **instance_type** [string] The EC2 instance type requested for the deployment.

    - **memory** [integer] The memory allocated to the deployment, in MB.

    - **cpu** [integer] The cpu allocated to the deployment, in millicores.

    - **state** [string] The state of the deployment.

    - **state_message** [string] A detailed description of the state.

- **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.

- **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.

- created_at : string/time

- updated_at : string/time

- **notebook_id** [integer] The ID of owning Notebook

- **credentials** [list] A list of credential IDs to pass to the notebook.
- **environment_variables** [dict] Environment variables to be passed into the Notebook.
- **idle_timeout** [integer] How long the notebook will stay alive without any kernel activity.
- **partition_label** [string] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.
- **git_repo_id** [integer] The ID of the git repository.
- **git_repo_url** [string] The url of the git repository
- **git_ref** [string] The git reference if git repo is specified
- **git_path** [string] The path to the .ipynb file in the git repo that will be started up on notebook launch
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.

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

civis.response.Response

- **id** [integer] The ID for this notebook.
- **name** [string] The name of this notebook.
- **language** [string] The kernel language of this notebook.
- **description** [string] The description of this notebook.
- **notebook_url** [string] Time-limited URL to get the .ipynb file for this notebook.
- **notebook_preview_url** [string] Time-limited URL to get the .htm preview file for this notebook.
- **requirements_url** [string] Time-limited URL to get the requirements.txt file for this notebook.
- **file_id** [string] The file ID for the S3 file containing the .ipynb file.
- **requirements_file_id** [string] The file ID for the S3 file containing the requirements.txt file.
- **user** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.

- **docker_image_name** [string] The name of the docker image to pull from DockerHub.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
- **instance_type** [string] The EC2 instance type to deploy to.
- **memory** [integer] The amount of memory allocated to the notebook.
- **cpu** [integer] The amount of cpu allocated to the the notebook.
- created_at : string/time
- updated_at : string/time
- **most_recent_deployment** [dict::]

  - **deployment_id** [integer] The ID for this deployment.

  - **user_id** [integer] The ID of the owner.

  - **host** [string] Domain of the deployment.

  - **name** [string] Name of the deployment.

  - **docker_image_name** [string] The name of the docker image to pull from DockerHub.

  - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).

  - **display_url** [string] A signed URL for viewing the deployed item.

  - **instance_type** [string] The EC2 instance type requested for the deployment.

  - **memory** [integer] The memory allocated to the deployment, in MB.

  - **cpu** [integer] The cpu allocated to the deployment, in millicores.

  - **state** [string] The state of the deployment.

  - **state_message** [string] A detailed description of the state.

  - **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.

  - **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.

  - created_at : string/time

  - updated_at : string/time

  - **notebook_id** [integer] The ID of owning Notebook
- **credentials** [list] A list of credential IDs to pass to the notebook.
- **environment_variables** [dict] Environment variables to be passed into the Notebook.
- **idle_timeout** [integer] How long the notebook will stay alive without any kernel activity.

- **partition_label** [string] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.
- **git_repo_id** [integer] The ID of the git repository.
- **git_repo_url** [string] The url of the git repository
- **git_ref** [string] The git reference if git repo is specified
- **git_path** [string] The path to the .ipynb file in the git repo that will be started up on notebook launch
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.

put_git(*id*, ***, *git_ref='DEFAULT'*, *git_branch='DEFAULT'*, *git_path='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref_type='DEFAULT'*, *pull_from_git='DEFAULT'*)

Attach an item to a file in a git repo

Parameters

id [integer] The ID of the file.

git_ref [string, optional] A git reference specifying an unambiguous version of the file. Can be a branch name, or the full or shortened SHA of a commit.

git_branch [string, optional] The git branch that the file is on.

git_path [string, optional] The path of the file in the repository.

git_repo_url [string, optional] The URL of the git repository.

git_ref_type [string, optional] Specifies if the file is versioned by branch or tag.

pull_from_git [boolean, optional] Automatically pull latest commit from git. Only works for scripts.

Returns

civis.response.Response

- **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
- **git_branch** [string] The git branch that the file is on.
- **git_path** [string] The path of the file in the repository.
- **git_repo** [dict::]

  – **id** [integer] The ID for this git repository.

  – **repo_url** [string] The URL for this git repository.

  – created_at : string/time

  – updated_at : string/time

- **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
- **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

put_projects(*id*, *project_id*)

Add a Notebook to a project

Parameters

id [integer] The ID of the Notebook.

project_id [integer] The ID of the project.

Returns

None Response code 204: success

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

    Set the permissions groups has on this object

        Parameters

            id [integer] The ID of the resource that is shared.

            group_ids [list] An array of one or more group IDs.

            permission_level [string] Options are: "read", "write", or "manage".

            share_email_body [string, optional] Custom body text for e-mail sent on a share.

            send_shared_email [boolean, optional] Send email to the recipients of a share.

        Returns

        civis.response.Response

- **readers** [dict::]

  - **users** [list::]

    * id : integer

    * name : string

  - **groups** [list::]

    * id : integer

    * name : string

- **writers** [dict::]

  - **users** [list::]

    * id : integer

    * name : string

  - **groups** [list::]

    * id : integer

    * name : string

- **owners** [dict::]

  - **users** [list::]

    * id : integer

    * name : string

  - **groups** [list::]

    * id : integer

    * name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

    Set the permissions users have on this object

        Parameters

            id [integer] The ID of the resource that is shared.

            user_ids [list] An array of one or more user IDs.

            permission_level [string] Options are: "read", "write", or "manage".

            share_email_body [string, optional] Custom body text for e-mail sent on a share.

send_shared_email [boolean, optional] Send email to the recipients of a share.

Returns

civis.response.Response

- **readers** [dict::]

  – **users** [list::]

    ∗ id : integer

    ∗ name : string

  – **groups** [list::]

    ∗ id : integer

    ∗ name : string

- **writers** [dict::]

  – **users** [list::]

    ∗ id : integer

    ∗ name : string

  – **groups** [list::]

    ∗ id : integer

    ∗ name : string

- **owners** [dict::]

  – **users** [list::]

    ∗ id : integer

    ∗ name : string

  – **groups** [list::]

    ∗ id : integer

    ∗ name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_transfer(*id*, *user_id*, *include_dependencies*, *, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

Transfer ownership of this object to another user

Parameters

id [integer] The ID of the resource that is shared.

user_id [integer] ID of target user

include_dependencies [boolean] Whether or not to give manage permissions on all dependencies

email_body [string, optional] Custom body text for e-mail sent on transfer.

send_email [boolean, optional] Send email to the target user of the transfer?

Returns

civis.response.Response

- **dependencies** [list::] Dependent objects for this object - object_type : string

  Dependent object type

  – **fco_type** [string] Human readable dependent object type

---

– **id** [integer] Dependent object ID

– **name** [string] Dependent object name, or nil if the requesting
user cannot read this object

– **permission_level** [string] Permission level of target user (not
user's groups) for dependent object, or null if no target
user

– **shared** [boolean] Whether dependent object was successfully
shared with target user

Notifications

class **Notifications**(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.notifications.list(...)
```

Methods

list(*[, last_event_id, r, mock])	Receive a stream of notifications as they come in

———

list(***, *last_event_id='DEFAULT'*, *r='DEFAULT'*, *mock='DEFAULT'*)
Receive a stream of notifications as they come in

> **Parameters**
>> **last_event_id** [string, optional] allows browser to keep track of last event fired
>> **r** [string, optional] specifies retry/reconnect timeout
>> **mock** [string, optional] used for testing
> **Returns**
>> **None** Response code 200: success

Ontology

class Ontology(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.ontology.list(...)
```

Methods

list(*[, subset])	List the ontology of column names Civis uses

—

list(***, *subset='DEFAULT'*)

> List the ontology of column names Civis uses
>> **Parameters**
>>> **subset** [string, optional] A subset of fields to return.
>> **Returns**
>>> *civis.response.Response*
>>>> • key : string
>>>> • title : string
>>>> • **desc** [string] A description of this field.
>>>> • aliases : list

Permission_Sets

class Permission_Sets(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.permission_sets.list(...)
```

Methods

delete_resources(id, name)	Delete a resource in a permission set
delete_resources_shares_groups(id, name, ...)	Revoke the permissions a group has on this object
delete_resources_shares_users(id, name, user_id)	Revoke the permissions a user has on this object
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 a Permission Set
get_resources(id, name)	Get a resource in a permission set
list(*[, archived, author, limit, page_num, ...])	List Permission Sets
list_dependencies(id, *[, user_id])	List dependent objects for this object
list_resources(id, *[, limit, page_num, ...])	List resources in a permission set
list_resources_shares(id, name)	List users and groups permissioned on this object
list_shares(id)	List users and groups permissioned on this object
list_users_permissions(id, user_id)	Get all permissions for a user, in this permission set
patch(id, *[, name, description])	Update some attributes of this Permission Set
patch_resources(id, name, *[, description])	Update a resource in a permission set
post(name, *[, description])	Create a Permission Set
post_resources(id, name, *[, description])	Create a resource in a permission set
put(id, name, *[, description])	Replace all attributes of this Permission Set
put_archive(id, status)	Update the archive status of this object
put_resources_shares_groups(id, name, ...[, ...])	Set the permissions groups has on this object
put_resources_shares_users(id, name, ...[, ...])	Set the permissions users have on this object
put_shares_groups(id, group_ids, ...[, ...])	Set the permissions groups has on this object
put_shares_users(id, user_ids, ...[, ...])	Set the permissions users have on this object
put_transfer(id, user_id, ...[, email_body, ...])	Transfer ownership of this object to another user

———

delete_resources(*id*, *name*)

> Delete a resource in a permission set
>> **Parameters**
>>> **id** [integer] The ID for this permission set.
>>> **name** [string] The name of this resource.
>> **Returns**
>>> **None** Response code 204: success

delete_resources_shares_groups(*id*, *name*, *group_id*)

> Revoke the permissions a group has on this object
>> **Parameters**
>>> **id** [integer] The ID for this permission set.
>>> **name** [string] The name of this resource.

> > > **group_id** [integer] The ID of the group.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

delete_resources_shares_users(*id*, *name*, *user_id*)

> Revoke the permissions a user has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID for this permission set.
> > >
> > > **name** [string] The name of this resource.
> > >
> > > **user_id** [integer] The ID of the user.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

delete_shares_groups(*id*, *group_id*)

> Revoke the permissions a group has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **group_id** [integer] The 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] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] The ID of the user.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

get(*id*)

> Get a Permission Set
>
> > **Parameters**
> >
> > > **id** [integer]
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > - **id** [integer] The ID for this permission set.
> > > > - **name** [string] The name of this permission set.
> > > > - **description** [string] A description of this permission set.
> > > > - **author** [dict::]
> > > >
> > > > > - **id** [integer] The ID of this user.
> > > > >
> > > > > - **name** [string] This user's name.
> > > > >
> > > > > - **username** [string] This user's username.
> > > > >
> > > > > - **initials** [string] This user's initials.
> > > > >
> > > > > - **online** [boolean] Whether this user is online.
> > > >
> > > > - created_at : string/time
> > > > - updated_at : string/time
> > > > - **archived** [string] The archival status of the requested item(s).

get_resources(*id*, *name*)

> Get a resource in a permission set
>
> > **Parameters**
> >
> > > **id** [integer] The ID for this permission set.
> > >
> > > **name** [string] The name of this resource.
> >
> > **Returns**

> *civis.response.Response*
>> • **permission_set_id** [integer] The ID for the permission set this resource belongs to.
>> • **name** [string] The name of this resource.
>> • **description** [string] A description of this resource.
>> • created_at : string/time
>> • updated_at : string/time

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

> List Permission Sets

> **Parameters**
>> **archived** [string, optional] The archival status of the requested item(s).
>> **author** [string, optional] If specified, return items from any of these authors. It accepts a comma- separated list of user IDs.
>> **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 50.
>> **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
>> **order** [string, optional] The field on which to order the result set. Defaults to updated_at. Must be one of: updated_at, name, created_at.
>> **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
>> **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

> **Returns**
> *civis.response.PaginatedResponse*
>> • **id** [integer] The ID for this permission set.
>> • **name** [string] The name of this permission set.
>> • **description** [string] A description of this permission set.
>> • **author** [dict::]
>>
>>> – **id** [integer] The ID of this user.
>>>
>>> – **name** [string] This user's name.
>>>
>>> – **username** [string] This user's username.
>>>
>>> – **initials** [string] This user's initials.
>>>
>>> – **online** [boolean] Whether this user is online.
>> • created_at : string/time
>> • updated_at : string/time
>> • **archived** [string] The archival status of the requested item(s).

list_dependencies(*id*, *, *user_id='DEFAULT'*)

> List dependent objects for this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>> **user_id** [integer, optional] ID of target user

> **Returns**
> *civis.response.Response*
>> • **object_type** [string] Dependent object type
>> • **fco_type** [string] Human readable dependent object type
>> • **id** [integer] Dependent object ID
>> • **name** [string] Dependent object name, or nil if the requesting user cannot

read this object

- **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
- **shareable** [boolean] Whether or not the requesting user can share this object.

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

List resources in a permission set

Parameters

id [integer] The ID for this permission set.

limit [integer, optional] Number of results to return. Defaults to 50. Maximum allowed is 1000.

page_num [integer, optional] Page number of the results to return. Defaults to the first page, 1.

order [string, optional] The field on which to order the result set. Defaults to name. Must be one of: name, id, updated_at, created_at.

order_dir [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to asc.

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

Returns

civis.response.PaginatedResponse

- **permission_set_id** [integer] The ID for the permission set this resource belongs to.
- **name** [string] The name of this resource.
- **description** [string] A description of this resource.
- created_at : string/time
- updated_at : string/time

list_resources_shares(*id*, *name*)

List users and groups permissioned on this object

Parameters

id [integer] The ID for this permission set.

name [string] The name of this resource.

Returns

civis.response.Response

- **readers** [dict::]
  - **users** [list::]
    * id : integer
    * name : string
  - **groups** [list::]
    * id : integer
    * name : string
- **writers** [dict::]
  - **users** [list::]
    * id : integer
    * name : string
  - **groups** [list::]

        ∗ id : integer

        ∗ name : string

- **owners** [dict::]

    - **users** [list::]

        ∗ id : integer

        ∗ name : string

    - **groups** [list::]

        ∗ id : integer

        ∗ name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_shares(*id*)

    List users and groups permissioned on this object

        Parameters

            id [integer] The ID of the resource that is shared.

        Returns

        civis.response.Response

- **readers** [dict::]

    - **users** [list::]

        ∗ id : integer

        ∗ name : string

    - **groups** [list::]

        ∗ id : integer

        ∗ name : string

- **writers** [dict::]

    - **users** [list::]

        ∗ id : integer

        ∗ name : string

    - **groups** [list::]

        ∗ id : integer

        ∗ name : string

- **owners** [dict::]

    - **users** [list::]

        ∗ id : integer

        ∗ name : string

    - **groups** [list::]

        ∗ id : integer

        ∗ name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_users_permissions(*id*, *user_id*)

Get all permissions for a user, in this permission set

Parameters

id [integer] The ID for this permission set.

user_id [integer] The ID for the user.

Returns

civis.response.Response

- **resource_name** [string] The name of the resource.
- **read** [boolean] If true, the user has read permission on this resource.
- **write** [boolean] If true, the user has write permission on this resource.
- **manage** [boolean] If true, the user has manage permission on this resource.

patch(*id*, ***, *name='DEFAULT'*, *description='DEFAULT'*)

Update some attributes of this Permission Set

Parameters

id [integer] The ID for this permission set.

name [string, optional] The name of this permission set.

description [string, optional] A description of this permission set.

Returns

civis.response.Response

- **id** [integer] The ID for this permission set.
- **name** [string] The name of this permission set.
- **description** [string] A description of this permission set.
- **author** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.
- created_at : string/time
- updated_at : string/time
- **archived** [string] The archival status of the requested item(s).

patch_resources(*id*, *name*, ***, *description='DEFAULT'*)

Update a resource in a permission set

Parameters

id [integer] The ID for this permission set.

name [string] The name of this resource.

description [string, optional] A description of this resource.

Returns

civis.response.Response

- **permission_set_id** [integer] The ID for the permission set this resource belongs to.
- **name** [string] The name of this resource.
- **description** [string] A description of this resource.
- created_at : string/time
- updated_at : string/time

post(*name*, ***, *description='DEFAULT'*)

> Create a Permission Set
>
> > **Parameters**
> >
> > > **name** [string] The name of this permission set.
> > >
> > > **description** [string, optional] A description of this permission set.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **id** [integer] The ID for this permission set.
> > > > • **name** [string] The name of this permission set.
> > > > • **description** [string] A description of this permission set.
> > > > • **author** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > • created_at : string/time
> > > > • updated_at : string/time
> > > > • **archived** [string] The archival status of the requested item(s).

post_resources(*id*, *name*, ***, *description='DEFAULT'*)

> Create a resource in a permission set
>
> > **Parameters**
> >
> > > **id** [integer] The ID for this permission set.
> > >
> > > **name** [string] The name of this resource.
> > >
> > > **description** [string, optional] A description of this resource.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **permission_set_id** [integer] The ID for the permission set this resource
> > > >   belongs to.
> > > > • **name** [string] The name of this resource.
> > > > • **description** [string] A description of this resource.
> > > > • created_at : string/time
> > > > • updated_at : string/time

put(*id*, *name*, ***, *description='DEFAULT'*)

> Replace all attributes of this Permission Set
>
> > **Parameters**
> >
> > > **id** [integer] The ID for this permission set.
> > >
> > > **name** [string] The name of this permission set.
> > >
> > > **description** [string, optional] A description of this permission set.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **id** [integer] The ID for this permission set.
> > > > • **name** [string] The name of this permission set.
> > > > • **description** [string] A description of this permission set.
> > > > • **author** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.

> > > > > – **initials** [string] This user's initials.
> > > >
> > > > – **online** [boolean] Whether this user is online.
> > >
> > > • created_at : string/time
> > > • updated_at : string/time
> > > • **archived** [string] The archival status of the requested item(s).

put_archive(*id*, *status*)

> Update the archive status of this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the object.
> > >
> > > **status** [boolean] The desired archived status of the object.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > > • **id** [integer] The ID for this permission set.
> > > > • **name** [string] The name of this permission set.
> > > > • **description** [string] A description of this permission set.
> > > > • **author** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > • created_at : string/time
> > > > • updated_at : string/time
> > > > • **archived** [string] The archival status of the requested item(s).

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

> Set the permissions groups has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID for this permission set.
> > >
> > > **name** [string] The name of this resource.
> > >
> > > **group_ids** [list] An array of one or more group IDs.
> > >
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > >
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > >
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > > • **readers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > >
> > > > • **writers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > * id : integer

> > > > > ∗ name : string

> > > > – **groups** [list::]

> > > > > ∗ id : integer

> > > > > ∗ name : string

> > > • **owners** [dict::]

> > > > – **users** [list::]

> > > > > ∗ id : integer

> > > > > ∗ name : string

> > > > – **groups** [list::]

> > > > > ∗ id : integer

> > > > > ∗ name : string

> > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object

> > **Parameters**

> > > **id** [integer] The ID for this permission set.
> > > **name** [string] The name of this resource.
> > > **user_ids** [list] An array of one or more user IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.

> > **Returns**

> > [*civis.response.Response*]

> > > • **readers** [dict::]

> > > > – **users** [list::]

> > > > > ∗ id : integer

> > > > > ∗ name : string

> > > > – **groups** [list::]

> > > > > ∗ id : integer

> > > > > ∗ name : string

> > > • **writers** [dict::]

> > > > – **users** [list::]

> > > > > ∗ id : integer

> > > > > ∗ name : string

> > > > – **groups** [list::]

> > > > > ∗ id : integer

> > > > > ∗ name : string

> > > • **owners** [dict::]

- **users** [list::]

    * id : integer

    * name : string

- **groups** [list::]

    * id : integer

    * name : string
- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions groups has on this object

> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > > **group_ids** [list] An array of one or more group IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > - **readers** [dict::]
> >
> >   - **users** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> >   - **groups** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> > - **writers** [dict::]
> >
> >   - **users** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> >   - **groups** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> > - **owners** [dict::]
> >
> >   - **users** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> >   - **groups** [list::]
> >
> >   * id : integer
> >
> >   * name : string

> • **total_user_shares** [integer] For owners, the number of total users shared.
> For writers and readers, the number of visible users shared.
> • **total_group_shares** [integer] For owners, the number of total groups
> shared. For writers and readers, the number of visible groups shared.

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

Set the permissions users have on this object

> **Parameters**
> **id** [integer] The ID of the resource that is shared.
> **user_ids** [list] An array of one or more user IDs.
> **permission_level** [string] Options are: "read", "write", or "manage".
> **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> **send_shared_email** [boolean, optional] Send email to the recipients of a share.
>
> **Returns**
> [*civis.response.Response*]
> • **readers** [dict::]
>> – **users** [list::]
>>> * id : integer
>>> * name : string
>> – **groups** [list::]
>>> * id : integer
>>> * name : string
> • **writers** [dict::]
>> – **users** [list::]
>>> * id : integer
>>> * name : string
>> – **groups** [list::]
>>> * id : integer
>>> * name : string
> • **owners** [dict::]
>> – **users** [list::]
>>> * id : integer
>>> * name : string
>> – **groups** [list::]
>>> * id : integer
>>> * name : string
> • **total_user_shares** [integer] For owners, the number of total users shared.
> For writers and readers, the number of visible users shared.
> • **total_group_shares** [integer] For owners, the number of total groups
> shared. For writers and readers, the number of visible groups shared.

put_transfer(*id*, *user_id*, *include_dependencies*, *, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

Transfer ownership of this object to another user

> **Parameters**
> **id** [integer] The ID of the resource that is shared.

> **user_id** [integer] ID of target user
>
> **include_dependencies** [boolean] Whether or not to give manage permissions on all dependencies
>
> **email_body** [string, optional] Custom body text for e-mail sent on transfer.
>
> **send_email** [boolean, optional] Send email to the target user of the transfer?

> **Returns**
>
> *civis.response.Response*
>
> > • **dependencies** [list::] Dependent objects for this object - object_type : string
> >
> > > Dependent object type
> >
> > > – **fco_type** [string] Human readable dependent object type
> > >
> > > – **id** [integer] Dependent object ID
> > >
> > > – **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > >
> > > – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > >
> > > – **shared** [boolean] Whether dependent object was successfully shared with target user

Predictions

class Predictions(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.predictions.list(...)
```

Methods

get(id)	Show the specified prediction
list(*[, model_id])	List predictions
list_schedules(id)	Show the prediction schedule

———

get(*id*)

> Show the specified prediction
>
> > **Parameters**
> >
> > **id** [integer] The ID of the prediction.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > • **id** [integer] The ID of the prediction.

- **model_id** [integer] The ID of the model used for this prediction.
- **scored_table_id** [integer] The ID of the source table for this prediction.
- **scored_table_name** [string] The name of the source table for this prediction.
- **output_table_name** [string] The name of the output table for this prediction.
- **state** [string] The state of the last run of this prediction.
- **error** [string] The error, if any, of the last run of this prediction.
- **started_at** [string/date-time] The start time of the last run of this prediction.
- **finished_at** [string/date-time] The end time of the last run of this prediction.
- **last_run** [dict::]

    – id : integer

    – state : string

    – **created_at** [string/time] The time that the run was queued.

    – **started_at** [string/time] The time that the run started.

    – **finished_at** [string/time] The time that the run completed.

    – **error** [string] The error message for this run, if present.

- **scored_tables** [list::] An array of created prediction tables. - id : integer

    The ID of the table with created predictions.

    – **schema** [string] The schema of table with created predictions.

    – **name** [string] The name of table with created predictions.

    – **created_at** [string/date-time] The time when the table with created predictions was created.

    – **score_stats** [list::] An array of metrics on the created predictions. - score_name : string

        The name of the score.

        * **histogram** [list] The histogram of the distribution of scores.

        * **avg_score** [number/float] The average score.

        * **min_score** [number/float] The minimum score.

        * **max_score** [number/float] The maximum score.

- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

    – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    – **scheduled_hours** [list] Hours of the day it is scheduled on.

    – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

> > – **scheduled_runs_per_hour** [integer] Alternative to sched-
> > uled minutes, number of times to run per hour.
>
> > – **scheduled_days_of_month** [list] Days of the month it is
> > scheduled on, mutually exclusive with scheduledDays.
> - **limiting_sql** [string] A SQL WHERE clause used to scope the rows to be
>   predicted.
> - **primary_key** [list] The primary key or composite keys of the table being
>   predicted.

list(*, *model_id='DEFAULT'*)

> List predictions
> > **Parameters**
> > > **model_id** [integer, optional] If specified, only return predictions associated with this
> > > model ID.
> >
> > **Returns**
> > > *civis.response.Response*
> > > > - **id** [integer] The ID of the prediction.
> > > > - **model_id** [integer] The ID of the model used for this prediction.
> > > > - **scored_table_id** [integer] The ID of the source table for this prediction.
> > > > - **scored_table_name** [string] The name of the source table for this predic-
> > > >   tion.
> > > > - **output_table_name** [string] The name of the output table for this predic-
> > > >   tion.
> > > > - **state** [string] The state of the last run of this prediction.
> > > > - **error** [string] The error, if any, of the last run of this prediction.
> > > > - **started_at** [string/date-time] The start time of the last run of this predic-
> > > >   tion.
> > > > - **finished_at** [string/date-time] The end time of the last run of this predic-
> > > >   tion.
> > > > - **last_run** [dict::]
> > > >
> > > > > – id : integer
> > > > >
> > > > > – state : string
> > > > >
> > > > > – **created_at** [string/time] The time that the run was queued.
> > > > >
> > > > > – **started_at** [string/time] The time that the run started.
> > > > >
> > > > > – **finished_at** [string/time] The time that the run completed.
> > > > >
> > > > > – **error** [string] The error message for this run, if present.

list_schedules(*id*)

> Show the prediction schedule
> > **Parameters**
> > > **id** [integer] ID of the prediction associated with this schedule.
> >
> > **Returns**
> > > *civis.response.Response*
> > > > - **id** [integer] ID of the prediction associated with this schedule.
> > > > - **schedule** [dict::]
> > > >
> > > > > – **scheduled** [boolean] If the item is scheduled.
> > > > >
> > > > > – **scheduled_days** [list] Days of the week, based on numeric
> > > > > value starting at 0 for Sunday. Mutually exclusive with
> > > > > scheduledDaysOfMonth
> > > > >
> > > > > – **scheduled_hours** [list] Hours of the day it is scheduled on.

– **scheduled_minutes** [list] Minutes of the day it is scheduled on.

– **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

– **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

• **score_on_model_build** [boolean] Whether the prediction will run after a rebuild of the associated model.

Projects

class Projects(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.projects.list(...)
```

Methods

delete_parent_projects(id, parent_project_id)	Remove an item from a Parent 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(project_id)	Get a detailed view of a project and the objects in it
list(*[, permission, author, hidden, ...])	List projects
list_dependencies(id, *[, user_id])	List dependent objects for this object
list_parent_projects(id, *[, hidden])	List the Parent Projects an item belongs to
list_shares(id)	List users and groups permissioned on this object
post(name, description, *[, note, hidden])	Create a project
put(project_id, *[, name, description, ...])	Update a project
put_archive(id, status)	Update the archive status of this object
put_parent_projects(id, parent_project_id)	Add an item to a Parent Project
put_shares_groups(id, group_ids, ...[, ...])	Set the permissions groups has on this object
put_shares_users(id, user_ids, ...[, ...])	Set the permissions users have on this object
put_transfer(id, user_id, ...[, email_body, ...])	Transfer ownership of this object to another user

———

delete_parent_projects(*id*, *parent_project_id*)

Remove an item from a Parent Project

Parameters

id [integer] The ID of the item.

parent_project_id [integer] The ID of the Parent 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] The ID of the resource that is shared.
> > >
> > > **group_id** [integer] The 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] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] The ID of the user.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

get(*project_id*)

> Get a detailed view of a project and the objects in it
>
> > **Parameters**
> >
> > > **project_id** [integer]
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > - **id** [integer] The ID for this project.
> > > > - **author** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > > - **name** [string] The name of this project.
> > > > - **description** [string] A description of the project.
> > > > - **users** [list::] Users who can see the project. - id : integer
> > > >
> > > > > The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > > - auto_share : boolean
> > > > - created_at : string/time
> > > > - updated_at : string/time
> > > > - **tables** [list::]
> > > >
> > > > > – schema : string
> > > > >
> > > > > – name : string
> > > > >
> > > > > – row_count : integer
> > > > >
> > > > > – column_count : integer
> > > > >
> > > > > – created_at : string/time

* **–** updated_at : string/time
* **surveys** [list::]

    * **– id** [integer] The item's ID.

    * **–** created_at : string/time

    * **–** updated_at : string/time
* **scripts** [list::]

    * **– id** [integer] The item's ID.

    * **–** created_at : string/time

    * **–** updated_at : string/time

    * **–** name : string

    * **–** type : string

    * **–** finished_at : string/time

    * **–** state : string

    * **– last_run** [dict::]

        * ∗ state : string

        * ∗ updated_at : string/time
* **imports** [list::]

    * **– id** [integer] The item's ID.

    * **–** created_at : string/time

    * **–** updated_at : string/time

    * **–** name : string

    * **–** type : string

    * **–** finished_at : string/time

    * **–** state : string

    * **– last_run** [dict::]

        * ∗ state : string

        * ∗ updated_at : string/time
* **exports** [list::]

    * **– id** [integer] The item's ID.

    * **–** created_at : string/time

    * **–** updated_at : string/time

    * **–** name : string

    * **–** type : string

    * **–** finished_at : string/time

    * **–** state : string

    * **– last_run** [dict::]

        * ∗ state : string

Civis Client Documentation, Release 1.16.0

* updated_at : string/time
- **models** [list::]
    - **id** [integer] The item's ID.
    - created_at : string/time
    - updated_at : string/time
    - name : string
    - state : string
- **notebooks** [list::]
    - **id** [integer] The item's ID.
    - created_at : string/time
    - updated_at : string/time
    - name : string
    - current_deployment_id : integer
    - **last_deploy** [dict::]
        * state : string
        * updated_at : string/time
- **services** [list::]
    - **id** [integer] The item's ID.
    - created_at : string/time
    - updated_at : string/time
    - name : string
    - current_deployment_id : integer
    - **last_deploy** [dict::]
        * state : string
        * updated_at : string/time
- **workflows** [list::]
    - **id** [integer] The item's ID.
    - created_at : string/time
    - updated_at : string/time
    - name : string
    - state : string
    - **last_execution** [dict::]
        * state : string
        * updated_at : string/time
- **reports** [list::]
    - **id** [integer] The item's ID.
    - created_at : string/time
    - updated_at : string/time

- **– name : string**

- **– state : string**

- **script_templates** [list::]

  - **– id** [integer] The item's ID.

  - **– created_at : string/time**

  - **– updated_at : string/time**

  - **– name : string**

- **files** [list::]

  - **– id** [integer] The item's ID.

  - **– created_at : string/time**

  - **– updated_at : string/time**

  - **– file_name : string**

  - **– file_size : integer**

  - **– expired : boolean**

- **enhancements** [list::]

  - **– id** [integer] The item's ID.

  - **– created_at : string/time**

  - **– updated_at : string/time**

  - **– name : string**

  - **– last_run** [dict::]

    - ∗ state : string

    - ∗ updated_at : string/time

- **projects** [list::]

  - **– id** [integer] The item's ID.

  - **– created_at : string/time**

  - **– updated_at : string/time**

  - **– name : string**

  - **– description : string**

- **all_objects** [list::]

  - **– project_id : integer**

  - **– object_id : integer**

  - **– object_type : string**

  - **– fco_type : string**

  - **– sub_type : string**

  - **– name : string**

  - **– icon : string**

  - **– author : string**

  - **– updated_at : string/time**

> > > – **archived** [string] The archival status of the requested item(s).
>
> > > – **hidden** [boolean] The hidden status of the item.
>
> > - note : string
> > - **hidden** [boolean] The hidden status of the item.
> > - **archived** [string] The archival status of the requested item(s).
> > - **parent_project** [dict::]
>
> > > – **id** [integer] The parent project's ID.
>
> > > – **name** [integer] The parent project's name.

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

> List projects
>
> > **Parameters**
> >
> > > **permission** [string, optional] A permissions string, one of "read", "write", or "manage". Lists only projects for which the current user has that permission.
> > >
> > > **author** [string, optional] If specified, return items from any of these authors. It accepts a comma- separated list of user IDs.
> > >
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> > >
> > > **archived** [string, optional] The archival status of the requested item(s).
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 1000.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to updated_at. Must be one of: updated_at, name, created_at.
> > >
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > >
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> >
> > **Returns**
> >
> > > [*civis.response.PaginatedResponse*](#)
> > >
> > > > - **id** [integer] The ID for this project.
> > > > - **author** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > - **name** [string] The name of this project.
> > > > - **description** [string] A description of the project.
> > > > - **users** [list::] Users who can see the project. - id : integer
> > > >
> > > > > The ID of this user.
> > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.

> > > > > – **online** [boolean] Whether this user is online.
> > > > - auto_share : boolean
> > > > - created_at : string/time
> > > > - updated_at : string/time
> > > > - **archived** [string] The archival status of the requested item(s).

list_dependencies(*id*, ***, *user_id='DEFAULT'*)

> List dependent objects for this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **user_id** [integer, optional] ID of target user
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **object_type** [string] Dependent object type
> > > > - **fco_type** [string] Human readable dependent object type
> > > > - **id** [integer] Dependent object ID
> > > > - **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > > > - **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > > > - **shareable** [boolean] Whether or not the requesting user can share this object.

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

> List the Parent Projects an item belongs to
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the item.
> > >
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **id** [integer] The ID for this project.
> > > > - **author** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > - **name** [string] The name of this project.
> > > > - **description** [string] A description of the project.
> > > > - **users** [list::] Users who can see the project. - id : integer
> > > >
> > > > > The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > - auto_share : boolean
> > > > - created_at : string/time

- updated_at : string/time
- **archived** [string] The archival status of the requested item(s).

list_shares(*id*)

    List users and groups permissioned on this object

        Parameters

            id [integer] The ID of the resource that is shared.

        Returns

        [*civis.response.Response*](#)

- **readers** [dict::]

    – **users** [list::]

        * id : integer

        * name : string

    – **groups** [list::]

        * id : integer

        * name : string

- **writers** [dict::]

    – **users** [list::]

        * id : integer

        * name : string

    – **groups** [list::]

        * id : integer

        * name : string

- **owners** [dict::]

    – **users** [list::]

        * id : integer

        * name : string

    – **groups** [list::]

        * id : integer

        * name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

post(*name*, *description*, ***, *note='DEFAULT'*, *hidden='DEFAULT'*)

    Create a project

        Parameters

            name [string] The name of this project.

            description [string] A description of the project.

            note [string, optional] Notes for the project.

            hidden [boolean, optional] The hidden status of the item.

        Returns

        [*civis.response.Response*](#)

- **id** [integer] The ID for this project.
- **author** [dict::]

- **id** [integer] The ID of this user.

- **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **name** [string] The name of this project.
- **description** [string] A description of the project.
- **users** [list::] Users who can see the project. - id : integer

    The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- auto_share : boolean
- created_at : string/time
- updated_at : string/time
- **tables** [list::]

    - schema : string

    - name : string

    - row_count : integer

    - column_count : integer

    - created_at : string/time

    - updated_at : string/time
- **surveys** [list::]

    - **id** [integer] The item's ID.

    - created_at : string/time

    - updated_at : string/time
- **scripts** [list::]

    - **id** [integer] The item's ID.

    - created_at : string/time

    - updated_at : string/time

    - name : string

    - type : string

    - finished_at : string/time

    - state : string

    - **last_run** [dict::]

        * state : string

        * updated_at : string/time
- **imports** [list::]

- **id** [integer] The item's ID.

- created_at : string/time

- updated_at : string/time

- name : string

- type : string

- finished_at : string/time

- state : string

- **last_run** [dict::]

  * state : string

  * updated_at : string/time

- **exports** [list::]

  - **id** [integer] The item's ID.

  - created_at : string/time

  - updated_at : string/time

  - name : string

  - type : string

  - finished_at : string/time

  - state : string

  - **last_run** [dict::]

    * state : string

    * updated_at : string/time

- **models** [list::]

  - **id** [integer] The item's ID.

  - created_at : string/time

  - updated_at : string/time

  - name : string

  - state : string

- **notebooks** [list::]

  - **id** [integer] The item's ID.

  - created_at : string/time

  - updated_at : string/time

  - name : string

  - current_deployment_id : integer

  - **last_deploy** [dict::]

    * state : string

    * updated_at : string/time

- **services** [list::]

- **id** [integer] The item's ID.

- created_at : string/time

- updated_at : string/time

- name : string

- current_deployment_id : integer

- **last_deploy** [dict::]

  * state : string

  * updated_at : string/time

- **workflows** [list::]

  - **id** [integer] The item's ID.

  - created_at : string/time

  - updated_at : string/time

  - name : string

  - state : string

  - **last_execution** [dict::]

    * state : string

    * updated_at : string/time

- **reports** [list::]

  - **id** [integer] The item's ID.

  - created_at : string/time

  - updated_at : string/time

  - name : string

  - state : string

- **script_templates** [list::]

  - **id** [integer] The item's ID.

  - created_at : string/time

  - updated_at : string/time

  - name : string

- **files** [list::]

  - **id** [integer] The item's ID.

  - created_at : string/time

  - updated_at : string/time

  - file_name : string

  - file_size : integer

  - expired : boolean

- **enhancements** [list::]

  - **id** [integer] The item's ID.

  - created_at : string/time

> - – updated_at : string/time
>
> - – name : string
>
> - – **last_run** [dict::]
>
>   - * state : string
>
>   - * updated_at : string/time
>
> - **projects** [list::]
>
>   - – **id** [integer] The item's ID.
>
>   - – created_at : string/time
>
>   - – updated_at : string/time
>
>   - – name : string
>
>   - – description : string
>
> - **all_objects** [list::]
>
>   - – project_id : integer
>
>   - – object_id : integer
>
>   - – object_type : string
>
>   - – fco_type : string
>
>   - – sub_type : string
>
>   - – name : string
>
>   - – icon : string
>
>   - – author : string
>
>   - – updated_at : string/time
>
>   - – **archived** [string] The archival status of the requested item(s).
>
>   - – **hidden** [boolean] The hidden status of the item.
>
> - note : string
> - **hidden** [boolean] The hidden status of the item.
> - **archived** [string] The archival status of the requested item(s).
> - **parent_project** [dict::]
>
>   - – **id** [integer] The parent project's ID.
>
>   - – **name** [integer] The parent project's name.

put(*project_id*, *, *name='DEFAULT'*, *description='DEFAULT'*, *note='DEFAULT'*, *auto_share='DEFAULT'*)

> Update a project
>> **Parameters**
>>> **project_id** [integer]
>>> **name** [string, optional] The name of this project.
>>> **description** [string, optional] A description of the project.
>>> **note** [string, optional] Notes for the project.
>>> **auto_share** [boolean, optional] A toggle for sharing the objects within the project
>>>> when the project is shared.This does not automatically share new objects to the
>>>> project.
>> **Returns**
>>> *civis.response.Response*
>>>> - **id** [integer] The ID for this project.

- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **name** [string] The name of this project.
- **description** [string] A description of the project.
- **users** [list::] Users who can see the project. - id : integer

    The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- auto_share : boolean
- created_at : string/time
- updated_at : string/time
- **tables** [list::]

    - schema : string

    - name : string

    - row_count : integer

    - column_count : integer

    - created_at : string/time

    - updated_at : string/time
- **surveys** [list::]

    - **id** [integer] The item's ID.

    - created_at : string/time

    - updated_at : string/time
- **scripts** [list::]

    - **id** [integer] The item's ID.

    - created_at : string/time

    - updated_at : string/time

    - name : string

    - type : string

    - finished_at : string/time

    - state : string

    - **last_run** [dict::]

        * state : string

> * updated_at : string/time

- **imports** [list::]

    - **id** [integer] The item's ID.

    - created_at : string/time

    - updated_at : string/time

    - name : string

    - type : string

    - finished_at : string/time

    - state : string

    - **last_run** [dict::]

        * state : string

        * updated_at : string/time

- **exports** [list::]

    - **id** [integer] The item's ID.

    - created_at : string/time

    - updated_at : string/time

    - name : string

    - type : string

    - finished_at : string/time

    - state : string

    - **last_run** [dict::]

        * state : string

        * updated_at : string/time

- **models** [list::]

    - **id** [integer] The item's ID.

    - created_at : string/time

    - updated_at : string/time

    - name : string

    - state : string

- **notebooks** [list::]

    - **id** [integer] The item's ID.

    - created_at : string/time

    - updated_at : string/time

    - name : string

    - current_deployment_id : integer

    - **last_deploy** [dict::]

        * state : string

∗ updated_at : string/time

- **services** [list::]

  – **id** [integer] The item's ID.

  – created_at : string/time

  – updated_at : string/time

  – name : string

  – current_deployment_id : integer

  – **last_deploy** [dict::]

    ∗ state : string

    ∗ updated_at : string/time

- **workflows** [list::]

  – **id** [integer] The item's ID.

  – created_at : string/time

  – updated_at : string/time

  – name : string

  – state : string

  – **last_execution** [dict::]

    ∗ state : string

    ∗ updated_at : string/time

- **reports** [list::]

  – **id** [integer] The item's ID.

  – created_at : string/time

  – updated_at : string/time

  – name : string

  – state : string

- **script_templates** [list::]

  – **id** [integer] The item's ID.

  – created_at : string/time

  – updated_at : string/time

  – name : string

- **files** [list::]

  – **id** [integer] The item's ID.

  – created_at : string/time

  – updated_at : string/time

  – file_name : string

  – file_size : integer

  – expired : boolean

- **enhancements** [list::]

- **id** [integer] The item's ID.

- created_at : string/time

- updated_at : string/time

- name : string

- **last_run** [dict::]

    * state : string

    * updated_at : string/time

- **projects** [list::]

    - **id** [integer] The item's ID.

    - created_at : string/time

    - updated_at : string/time

    - name : string

    - description : string

- **all_objects** [list::]

    - project_id : integer

    - object_id : integer

    - object_type : string

    - fco_type : string

    - sub_type : string

    - name : string

    - icon : string

    - author : string

    - updated_at : string/time

    - **archived** [string] The archival status of the requested item(s).

    - **hidden** [boolean] The hidden status of the item.

- note : string
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **parent_project** [dict::]

    - **id** [integer] The parent project's ID.

    - **name** [integer] The parent project'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

    [*civis.response.Response*](#)

- **id** [integer] The ID for this project.
- **author** [dict::]

    - **id** [integer] The ID of this user.

- **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **name** [string] The name of this project.
- **description** [string] A description of the project.
- **users** [list::] Users who can see the project. - id : integer

    The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- auto_share : boolean
- created_at : string/time
- updated_at : string/time
- **tables** [list::]

    - schema : string

    - name : string

    - row_count : integer

    - column_count : integer

    - created_at : string/time

    - updated_at : string/time
- **surveys** [list::]

    - **id** [integer] The item's ID.

    - created_at : string/time

    - updated_at : string/time
- **scripts** [list::]

    - **id** [integer] The item's ID.

    - created_at : string/time

    - updated_at : string/time

    - name : string

    - type : string

    - finished_at : string/time

    - state : string

    - **last_run** [dict::]

        * state : string

        * updated_at : string/time
- **imports** [list::]

    - **id** [integer] The item's ID.

> > > > – created_at : string/time
> > > >
> > > > – updated_at : string/time
> > > >
> > > > – name : string
> > > >
> > > > – type : string
> > > >
> > > > – finished_at : string/time
> > > >
> > > > – state : string
> > > >
> > > > – **last_run** [dict::]
> > > >
> > > > > > ∗ state : string
> > > > > >
> > > > > > ∗ updated_at : string/time

> > > • **exports** [list::]
> > >
> > > > > – **id** [integer] The item's ID.
> > > > >
> > > > > – created_at : string/time
> > > > >
> > > > > – updated_at : string/time
> > > > >
> > > > > – name : string
> > > > >
> > > > > – type : string
> > > > >
> > > > > – finished_at : string/time
> > > > >
> > > > > – state : string
> > > > >
> > > > > – **last_run** [dict::]
> > > > >
> > > > > > > ∗ state : string
> > > > > > >
> > > > > > > ∗ updated_at : string/time

> > > • **models** [list::]
> > >
> > > > > – **id** [integer] The item's ID.
> > > > >
> > > > > – created_at : string/time
> > > > >
> > > > > – updated_at : string/time
> > > > >
> > > > > – name : string
> > > > >
> > > > > – state : string

> > > • **notebooks** [list::]
> > >
> > > > > – **id** [integer] The item's ID.
> > > > >
> > > > > – created_at : string/time
> > > > >
> > > > > – updated_at : string/time
> > > > >
> > > > > – name : string
> > > > >
> > > > > – current_deployment_id : integer
> > > > >
> > > > > – **last_deploy** [dict::]
> > > > >
> > > > > > > ∗ state : string
> > > > > > >
> > > > > > > ∗ updated_at : string/time

> > > • **services** [list::]
> > >
> > > > > – **id** [integer] The item's ID.

> > > > – created_at : string/time
> > > >
> > > > – updated_at : string/time
> > > >
> > > > – name : string
> > > >
> > > > – current_deployment_id : integer
> > > >
> > > > – **last_deploy** [dict::]
> > > >
> > > > > > ∗ state : string
> > > > > >
> > > > > > ∗ updated_at : string/time

> > • **workflows** [list::]
> >
> > > > – **id** [integer] The item's ID.
> > > >
> > > > – created_at : string/time
> > > >
> > > > – updated_at : string/time
> > > >
> > > > – name : string
> > > >
> > > > – state : string
> > > >
> > > > – **last_execution** [dict::]
> > > >
> > > > > > ∗ state : string
> > > > > >
> > > > > > ∗ updated_at : string/time

> > • **reports** [list::]
> >
> > > > – **id** [integer] The item's ID.
> > > >
> > > > – created_at : string/time
> > > >
> > > > – updated_at : string/time
> > > >
> > > > – name : string
> > > >
> > > > – state : string

> > • **script_templates** [list::]
> >
> > > > – **id** [integer] The item's ID.
> > > >
> > > > – created_at : string/time
> > > >
> > > > – updated_at : string/time
> > > >
> > > > – name : string

> > • **files** [list::]
> >
> > > > – **id** [integer] The item's ID.
> > > >
> > > > – created_at : string/time
> > > >
> > > > – updated_at : string/time
> > > >
> > > > – file_name : string
> > > >
> > > > – file_size : integer
> > > >
> > > > – expired : boolean

> > • **enhancements** [list::]
> >
> > > > – **id** [integer] The item's ID.
> > > >
> > > > – created_at : string/time
> > > >
> > > > – updated_at : string/time

> > > > > – name : string
> > > > >
> > > > > – **last_run** [dict::]
> > > > >
> > > > > > > * state : string
> > > > > >
> > > > > > > * updated_at : string/time

> > > > • **projects** [list::]
> > > >
> > > > > – **id** [integer] The item's ID.
> > > > >
> > > > > – created_at : string/time
> > > > >
> > > > > – updated_at : string/time
> > > > >
> > > > > – name : string
> > > > >
> > > > > – description : string

> > > > • **all_objects** [list::]
> > > >
> > > > > – project_id : integer
> > > > >
> > > > > – object_id : integer
> > > > >
> > > > > – object_type : string
> > > > >
> > > > > – fco_type : string
> > > > >
> > > > > – sub_type : string
> > > > >
> > > > > – name : string
> > > > >
> > > > > – icon : string
> > > > >
> > > > > – author : string
> > > > >
> > > > > – updated_at : string/time
> > > > >
> > > > > – **archived** [string] The archival status of the requested item(s).
> > > > >
> > > > > – **hidden** [boolean] The hidden status of the item.

> > > • note : string
> > > • **hidden** [boolean] The hidden status of the item.
> > > • **archived** [string] The archival status of the requested item(s).
> > > • **parent_project** [dict::]
> > >
> > > > – **id** [integer] The parent project's ID.
> > > >
> > > > – **name** [integer] The parent project's name.

put_parent_projects(*id*, *parent_project_id*)

> Add an item to a Parent Project
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the item.
> > >
> > > **parent_project_id** [integer] The ID of the Parent Project.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

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

> Set the permissions groups has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **group_ids** [list] An array of one or more group IDs.
> > >
> > > **permission_level** [string] Options are: "read", "write", or "manage".

> > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
>
> > **Returns**
> > > [*civis.response.Response*](#)
> > > > > * **readers** [dict::]
> > > > > > > – **users** [list::]
> > > > > > > > > ∗ id : integer
> > > > > > > > >
> > > > > > > > > ∗ name : string
> > > > > > >
> > > > > > > – **groups** [list::]
> > > > > > > > > ∗ id : integer
> > > > > > > > >
> > > > > > > > > ∗ name : string
> > > > >
> > > > > * **writers** [dict::]
> > > > > > > – **users** [list::]
> > > > > > > > > ∗ id : integer
> > > > > > > > >
> > > > > > > > > ∗ name : string
> > > > > > >
> > > > > > > – **groups** [list::]
> > > > > > > > > ∗ id : integer
> > > > > > > > >
> > > > > > > > > ∗ name : string
> > > > >
> > > > > * **owners** [dict::]
> > > > > > > – **users** [list::]
> > > > > > > > > ∗ id : integer
> > > > > > > > >
> > > > > > > > > ∗ name : string
> > > > > > >
> > > > > > > – **groups** [list::]
> > > > > > > > > ∗ id : integer
> > > > > > > > >
> > > > > > > > > ∗ name : string
> > > > >
> > > > > * **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > > * **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object
>
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_ids** [list] An array of one or more user IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> > > [*civis.response.Response*](#)
> > > > > * **readers** [dict::]
> > > > > > > – **users** [list::]
> > > > > > > > > ∗ id : integer

> > > > > > > > ∗ name : string

> > > > > > > – **groups** [list::]

> > > > > > > > > ∗ id : integer

> > > > > > > > > ∗ name : string

> > > > > > • **writers** [dict::]

> > > > > > > – **users** [list::]

> > > > > > > > > ∗ id : integer

> > > > > > > > > ∗ name : string

> > > > > > > – **groups** [list::]

> > > > > > > > > ∗ id : integer

> > > > > > > > > ∗ name : string

> > > > > > • **owners** [dict::]

> > > > > > > – **users** [list::]

> > > > > > > > > ∗ id : integer

> > > > > > > > > ∗ name : string

> > > > > > > – **groups** [list::]

> > > > > > > > > ∗ id : integer

> > > > > > > > > ∗ name : string

> > > > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

> > > > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_transfer(*id*, *user_id*, *include_dependencies*, *, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

> Transfer ownership of this object to another user

> > **Parameters**

> > > **id** [integer] The ID of the resource that is shared.

> > > **user_id** [integer] ID of target user

> > > **include_dependencies** [boolean] Whether or not to give manage permissions on all dependencies

> > > **email_body** [string, optional] Custom body text for e-mail sent on transfer.

> > > **send_email** [boolean, optional] Send email to the target user of the transfer?

> > **Returns**

> > > *civis.response.Response*

> > > > • **dependencies** [list::] Dependent objects for this object - object_type : string

> > > > > Dependent object type

> > > > > – **fco_type** [string] Human readable dependent object type

> > > > > – **id** [integer] Dependent object ID

> > > > > – **name** [string] Dependent object name, or nil if the requesting user cannot read this object

> > > > > – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user

> – **shared** [boolean] Whether dependent object was successfully
> shared with target user

Queries

class Queries(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.queries.list(...)
```

Methods

delete(id)	Sets Query Hidden to true
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(*[, database_id, credential_id, ...])	List
list_runs(id, *[, limit, page_num, order, ...])	List runs for the given query
list_runs_logs(id, run_id, *[, last_id, limit])	Get the logs for a run
post(database, sql, preview_rows, *[, ...])	Execute a query
post_runs(id)	Start a run
put_scripts(id, script_id)	Update the query's associated script

———

delete(*id*)
> Sets Query Hidden to true
> > **Parameters**
> > > **id** [integer] The query ID.
> > **Returns**
> > > *civis.response.Response*
> > > > - **id** [integer] The query ID.
> > > > - **database** [integer] The database ID.
> > > > - **sql** [string] The SQL to execute.
> > > > - **credential** [integer] The credential ID.
> > > > - **result_rows** [list] A preview of rows returned by the query.
> > > > - **result_columns** [list] A preview of columns returned by the query.
> > > > - **script_id** [integer] The ID of the script associated with this query.
> > > > - **exception** [string] Deprecated and not used.
> > > > - **error** [string] The error message for this run, if present.
> > > > - created_at : string/time
> > > > - updated_at : string/time
> > > > - **started_at** [string/date-time] The start time of the last run.
> > > > - **finished_at** [string/date-time] The end time of the last run.
> > > > - **state** [string] The state of the last run. One of queued, running, succeeded,
> > > > > failed, and cancelled.

- **last_run_id** [integer] The ID of the last run.
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **name** [string] The name of the query.
- **author** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.
- **report_id** [integer] The ID of the report associated with this query.

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**
>>> [*civis.response.Response*](#)
>>>> - **id** [integer] The query ID.
>>>> - **database** [integer] The database ID.
>>>> - **sql** [string] The SQL to execute.
>>>> - **credential** [integer] The credential ID.
>>>> - **result_rows** [list] A preview of rows returned by the query.
>>>> - **result_columns** [list] A preview of columns returned by the query.
>>>> - **script_id** [integer] The ID of the script associated with this query.
>>>> - **exception** [string] Deprecated and not used.
>>>> - **error** [string] The error message for this run, if present.
>>>> - created_at : string/time
>>>> - updated_at : string/time
>>>> - **started_at** [string/date-time] The start time of the last run.
>>>> - **finished_at** [string/date-time] The end time of the last run.
>>>> - **state** [string] The state of the last run. One of queued, running, succeeded, failed, and cancelled.
>>>> - **last_run_id** [integer] The ID of the last run.
>>>> - **hidden** [boolean] The hidden status of the item.
>>>> - **archived** [string] The archival status of the requested item(s).
>>>> - **name** [string] The name of the query.
>>>> - **author** [dict::]

>>>>> – **id** [integer] The ID of this user.

>>>>> – **name** [string] This user's name.

>>>>> – **username** [string] This user's username.

>>>>> – **initials** [string] This user's initials.

                 – **online** [boolean] Whether this user is online.
- **report_id** [integer] The ID of the report associated with this query.

get_runs(*id*, *run_id*)

    Check status of a run

        Parameters

            id [integer] The ID of the query.

            run_id [integer] The ID of the run.

        Returns

            civis.response.Response

- **id** [integer] The ID of the run.
- **query_id** [integer] The ID of the query.
- **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **is_cancel_requested** [boolean] True if run cancel requested, else false.
- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.
- **error** [string] The error, if any, returned by the run.

list(***, *database_id='DEFAULT'*, *credential_id='DEFAULT'*, *author_id='DEFAULT'*, *created_before='DEFAULT'*, *created_after='DEFAULT'*, *started_before='DEFAULT'*, *started_after='DEFAULT'*, *state='DEFAULT'*, *exclude_results='DEFAULT'*, *hidden='DEFAULT'*, *archived='DEFAULT'*, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)

    List

        Parameters

            database_id [integer, optional] The database ID.

            credential_id [integer, optional] The credential ID.

            author_id [integer, optional] The author of the query.

            created_before [string, optional] An upper bound for the creation date of the query.

            created_after [string, optional] A lower bound for the creation date of the query.

            started_before [string, optional] An upper bound for the start date of the last run.

            started_after [string, optional] A lower bound for the start date of the last run.

            state [array, optional] The state of the last run. One or more of queued, running, succeeded, failed, and cancelled. Specify multiple values as a comma-separated list (e.g., "A,B").

            exclude_results [boolean, optional] If true, does not return cached query results.

            hidden [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.

            archived [string, optional] The archival status of the requested item(s).

            limit [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 50.

            page_num [integer, optional] Page number of the results to return. Defaults to the first page, 1.

            order [string, optional] The field on which to order the result set. Defaults to created_at. Must be one of: created_at, started_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

            civis.response.PaginatedResponse

- **id** [integer] The query ID.
- **database** [integer] The database ID.

- **sql** [string] The SQL to execute.
- **credential** [integer] The credential ID.
- **result_rows** [list] A preview of rows returned by the query.
- **result_columns** [list] A preview of columns returned by the query.
- **script_id** [integer] The ID of the script associated with this query.
- **exception** [string] Deprecated and not used.
- **error** [string] The error message for this run, if present.
- created_at : string/time
- updated_at : string/time
- **started_at** [string/date-time] The start time of the last run.
- **finished_at** [string/date-time] The end time of the last run.
- **state** [string] The state of the last run. One of queued, running, succeeded, failed, and cancelled.
- **last_run_id** [integer] The ID of the last run.
- **archived** [string] The archival status of the requested item(s).
- **preview_rows** [integer] The number of rows to save from the query's result (maximum: 100).
- **report_id** [integer] The ID of the report associated with this query.

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

> List runs for the given query
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the query.
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 100.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to id. Must be one of: id.
> > >
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > >
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> >
> > **Returns**
> >
> > > *civis.response.PaginatedResponse*
> > >
> > > - **id** [integer] The ID of the run.
> > > - **query_id** [integer] The ID of the query.
> > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > - **started_at** [string/time] The time the last run started at.
> > > - **finished_at** [string/time] The time the last run completed.
> > > - **error** [string] The error, if any, returned by the run.

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

> Get the logs for a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the query.
> > >
> > > **run_id** [integer] The ID of the run.
> > >
> > > **last_id** [integer, optional] The ID of the last log message received. Log entries with this ID value or lower will be omitted.Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.
> > >
> > > **limit** [integer, optional] The maximum number of log messages to return. Default of

10000.

Returns

civis.response.Response

- **id** [integer] The ID of the log.
- **created_at** [string/date-time] The time the log was created.
- **message** [string] The log message.
- **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

post(*database*, *sql*, *preview_rows*, *, *credential='DEFAULT'*, *hidden='DEFAULT'*, *interactive='DEFAULT'*, *include_header='DEFAULT'*, *compression='DEFAULT'*, *column_delimiter='DEFAULT'*, *unquoted='DEFAULT'*, *filename_prefix='DEFAULT'*)

Execute a query

Parameters

database [integer] The database ID.

sql [string] The SQL to execute.

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

credential [integer, optional] The credential ID.

hidden [boolean, optional] The hidden status of the item.

interactive [boolean, optional] Deprecated and not used.

include_header [boolean, optional] Whether the CSV output should include a header row [default: true].

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

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

unquoted [boolean, optional] If true, will not quote fields.

filename_prefix [string, optional] The output filename prefix.

Returns

civis.response.Response

- **id** [integer] The query ID.
- **database** [integer] The database ID.
- **sql** [string] The SQL to execute.
- **credential** [integer] The credential ID.
- **result_rows** [list] A preview of rows returned by the query.
- **result_columns** [list] A preview of columns returned by the query.
- **script_id** [integer] The ID of the script associated with this query.
- **exception** [string] Deprecated and not used.
- **error** [string] The error message for this run, if present.
- created_at : string/time
- updated_at : string/time
- **started_at** [string/date-time] The start time of the last run.
- **finished_at** [string/date-time] The end time of the last run.
- **state** [string] The state of the last run. One of queued, running, succeeded, failed, and cancelled.
- **last_run_id** [integer] The ID of the last run.
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **interactive** [boolean] Deprecated and not used.
- **preview_rows** [integer] The number of rows to save from the query's result (maximum: 100).

- **include_header** [boolean] Whether the CSV output should include a header row [default: true].
- **compression** [string] The type of compression. One of gzip or zip, or none [default: gzip].
- **column_delimiter** [string] The delimiter to use. One of comma or tab, or pipe [default: comma].
- **unquoted** [boolean] If true, will not quote fields.
- **filename_prefix** [string] The output filename prefix.
- **report_id** [integer] The ID of the report associated with this query.

post_runs(*id*)

> Start a run
>> **Parameters**
>>> **id** [integer] The ID of the query.
>> **Returns**
>>> *civis.response.Response*
>>>> - **id** [integer] The ID of the run.
>>>> - **query_id** [integer] The ID of the query.
>>>> - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
>>>> - **is_cancel_requested** [boolean] True if run cancel requested, else false.
>>>> - **started_at** [string/time] The time the last run started at.
>>>> - **finished_at** [string/time] The time the last run completed.
>>>> - **error** [string] The error, if any, returned by the run.

put_scripts(*id*, *script_id*)

> Update the query's associated script
>> **Parameters**
>>> **id** [integer] The query ID.
>>> **script_id** [integer] The ID of the script associated with this query.
>> **Returns**
>>> *civis.response.Response*
>>>> - **id** [integer] The query ID.
>>>> - **database** [integer] The database ID.
>>>> - **sql** [string] The SQL to execute.
>>>> - **credential** [integer] The credential ID.
>>>> - **result_rows** [list] A preview of rows returned by the query.
>>>> - **result_columns** [list] A preview of columns returned by the query.
>>>> - **script_id** [integer] The ID of the script associated with this query.
>>>> - **exception** [string] Deprecated and not used.
>>>> - **error** [string] The error message for this run, if present.
>>>> - created_at : string/time
>>>> - updated_at : string/time
>>>> - **started_at** [string/date-time] The start time of the last run.
>>>> - **finished_at** [string/date-time] The end time of the last run.
>>>> - **state** [string] The state of the last run. One of queued, running, succeeded, failed, and cancelled.
>>>> - **last_run_id** [integer] The ID of the last run.
>>>> - **hidden** [boolean] The hidden status of the item.
>>>> - **archived** [string] The archival status of the requested item(s).
>>>> - **name** [string] The name of the query.
>>>> - **author** [dict::]
>>>>> – **id** [integer] The ID of this user.
>>>>>
>>>>> – **name** [string] This user's name.

– **username** [string] This user's username.

– **initials** [string] This user's initials.

– **online** [boolean] Whether this user is online.
- **report_id** [integer] The ID of the report associated with this query.

Remote_Hosts

class Remote_Hosts(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.remote_hosts.list_shares(...)
```

Methods

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
list(*[, type])	List the remote hosts
list_data_sets(id, *[, credential_id, ...])	List data sets available from a remote host
list_shares(id)	List users and groups permissioned on this object
post(name, url, type)	Create a new remote host
post_authenticate(id, *[, credential_id, ...])	Authenticate against a remote host using either a credential or a user name and password
put_shares_groups(id, group_ids, ...[, ...])	Set the permissions groups has on this object
put_shares_users(id, user_ids, ...[, ...])	Set the permissions users have on this object

—

delete_shares_groups(*id*, *group_id*)
> Revoke the permissions a group has on this object
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **group_id** [integer] The 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] The ID of the resource that is shared.
> > > **user_id** [integer] The ID of the user.
> > **Returns**
> > > **None** Response code 204: success

list(*, *type='DEFAULT'*)
> List the remote hosts
> > **Parameters**

> **type** [string, optional] The type of remote host. One of: RemoteHostTypes::Bigquery,
> RemoteHostTypes::Bitbucket, RemoteHostTypes::GitSSH, RemoteHost-
> Types::Github, RemoteHostTypes::GoogleDoc, RemoteHostTypes::JDBC,
> RemoteHostTypes::Postgres, RemoteHostTypes::Redshift, RemoteHost-
> Types::S3Storage, and RemoteHostTypes::Salesforce

> **Returns**
> > [*civis.response.Response*](#)
> > > • **id** [integer] The ID of the remote host.
> > > • **name** [string] The name of the remote host.
> > > • **type** [string] The type of remote host. One of: RemoteHost-
> > > > Types::Bigquery, RemoteHostTypes::Bitbucket, RemoteHost-
> > > > Types::GitSSH, RemoteHostTypes::Github, RemoteHost-
> > > > Types::GoogleDoc, RemoteHostTypes::JDBC, RemoteHost-
> > > > Types::Postgres, RemoteHostTypes::Redshift, RemoteHost-
> > > > Types::S3Storage, and RemoteHostTypes::Salesforce
> > > • **url** [string] The URL for remote host.

list_data_sets(*id*, ***, *credential_id='DEFAULT'*, *username='DEFAULT'*, *password='DEFAULT'*,
> > *q='DEFAULT'*, *s='DEFAULT'*)

> List data sets available from a remote host

> > **Parameters**
> > > **id** [integer] The ID of the remote host.
> > > **credential_id** [integer, optional] The credential ID.
> > > **username** [string, optional] The user name for remote host.
> > > **password** [string, optional] The password for remote host.
> > > **q** [string, optional] The query string for data set.
> > > **s** [boolean, optional] If true will only return schemas, otherwise, the results will be the
> > > > full path.

> > **Returns**
> > > [*civis.response.Response*](#)
> > > > • **name** [string] The path to a data_set.
> > > > • **full_path** [boolean] Boolean that indicates whether further querying
> > > > > needs to be done before the table can be selected.

list_shares(*id*)

> List users and groups permissioned on this object

> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.

> > **Returns**
> > > [*civis.response.Response*](#)
> > > > • **readers** [dict::]
> > > > > – **users** [list::]
> > > > > > ∗ id : integer
> > > > > > ∗ name : string
> > > > > – **groups** [list::]
> > > > > > ∗ id : integer
> > > > > > ∗ name : string
> > > > • **writers** [dict::]
> > > > > – **users** [list::]
> > > > > > ∗ id : integer

> > > > > > > > ∗ name : string
> > > > > > > – **groups** [list::]
> > > > > > > > ∗ id : integer
> > > > > > > > ∗ name : string
> > > > > > • **owners** [dict::]
> > > > > > > – **users** [list::]
> > > > > > > > ∗ id : integer
> > > > > > > > ∗ name : string
> > > > > > > – **groups** [list::]
> > > > > > > > ∗ id : integer
> > > > > > > > ∗ name : string
> > > > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

post(*name*, *url*, *type*)

> Create a new remote host
>
> > **Parameters**
> >
> > > **name** [string] The human readable name for the remote host.
> > >
> > > **url** [string] The URL to your host.
> > >
> > > **type** [string] The type of remote host. One of: RemoteHostTypes::Bigquery, RemoteHostTypes::Bitbucket, RemoteHostTypes::GitSSH, RemoteHostTypes::Github, RemoteHostTypes::GoogleDoc, RemoteHostTypes::JDBC, RemoteHostTypes::Postgres, RemoteHostTypes::Redshift, RemoteHostTypes::S3Storage, and RemoteHostTypes::Salesforce
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **id** [integer] The ID of the remote host.
> > > > • **name** [string] The name of the remote host.
> > > > • **type** [string] The type of remote host. One of: RemoteHostTypes::Bigquery, RemoteHostTypes::Bitbucket, RemoteHostTypes::GitSSH, RemoteHostTypes::Github, RemoteHostTypes::GoogleDoc, RemoteHostTypes::JDBC, RemoteHostTypes::Postgres, RemoteHostTypes::Redshift, RemoteHostTypes::S3Storage, and RemoteHostTypes::Salesforce
> > > > • **url** [string] The URL for remote host.

post_authenticate(*id*, ***, *credential_id='DEFAULT'*, *username='DEFAULT'*, *password='DEFAULT'*)

> Authenticate against a remote host using either a credential or a user name and password
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the remote host.
> > >
> > > **credential_id** [integer, optional] The credential ID.
> > >
> > > **username** [string, optional] The user name for remote host.
> > >
> > > **password** [string, optional] The password for remote host.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

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

> Set the permissions groups has on this object

Parameters

    id [integer] The ID of the resource that is shared.

    group_ids [list] An array of one or more group IDs.

    permission_level [string] Options are: "read", "write", or "manage".

    share_email_body [string, optional] Custom body text for e-mail sent on a share.

    send_shared_email [boolean, optional] Send email to the recipients of a share.

Returns

    civis.response.Response

- **readers** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **writers** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **owners** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

    Set the permissions users have on this object

    Parameters

        id [integer] The ID of the resource that is shared.

        user_ids [list] An array of one or more user IDs.

        permission_level [string] Options are: "read", "write", or "manage".

        share_email_body [string, optional] Custom body text for e-mail sent on a share.

        send_shared_email [boolean, optional] Send email to the recipients of a share.

    Returns

        civis.response.Response

- **readers** [dict::]
    - **users** [list::]
        - ∗ id : integer
        - ∗ name : string
    - **groups** [list::]
        - ∗ id : integer
        - ∗ name : string
- **writers** [dict::]
    - **users** [list::]
        - ∗ id : integer
        - ∗ name : string
    - **groups** [list::]
        - ∗ id : integer
        - ∗ name : string
- **owners** [dict::]
    - **users** [list::]
        - ∗ id : integer
        - ∗ name : string
    - **groups** [list::]
        - ∗ id : integer
        - ∗ name : string
- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Reports

class **Reports**(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.reports.list(...)
```

Methods

`delete_grants`(id)	Revoke permission for this report to perform Civis platform API operations on your behalf
`delete_projects`(id, project_id)	Remove a Report from a project
`delete_services_projects`(id, project_id)	Remove a Service Report from a project
`delete_services_shares_groups`(id, group_id)	Revoke the permissions a group has on this object
`delete_services_shares_users`(id, user_id)	Revoke the permissions a user has on this object
`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 a single report
`get_git_commits`(id, commit_hash)	Get file contents at git ref
`get_services`(id)	Show a single service report
`list`(*[, type, template_id, author, hidden, ...])	List Reports
`list_dependencies`(id, *[, user_id])	List dependent objects for this object
`list_git`(id)	Get the git metadata attached to an item
`list_git_commits`(id)	Get the git commits for an item on the current branch
`list_projects`(id, *[, hidden])	List the projects a Report belongs to
`list_services_dependencies`(id, *[, user_id])	List dependent objects for this object
`list_services_projects`(id, *[, hidden])	List the projects a Service Report belongs to
`list_services_shares`(id)	List users and groups permissioned on this object
`list_shares`(id)	List users and groups permissioned on this object
`patch`(id, *[, name, script_id, code_body, ...])	Update a report
`patch_git`(id, *[, git_ref, git_branch, ...])	Update an attached git file
`patch_services`(id, *[, name, provide_api_key])	Update some attributes of this service report
`post`(*[, script_id, name, code_body, ...])	Create a report
`post_git_checkout`(id)	Checkout content that the existing git_ref points to and save to the object
`post_git_checkout_latest`(id)	Checkout latest commit on the current branch of a script or workflow
`post_git_commits`(id, content, message, file_hash)	Commit and push a new version of the file
`post_grants`(id)	Grant this report the ability to perform Civis platform API operations on your behalf
`post_refresh`(id)	Refresh the data in this Tableau report
`post_services`(service_id, *[, provide_api_key])	Create a service report
`put_archive`(id, status)	Update the archive status of this object
`put_git`(id, *[, git_ref, git_branch, ...])	Attach an item to a file in a git repo
`put_projects`(id, project_id)	Add a Report to a project
`put_services_archive`(id, status)	Update the archive status of this object
`put_services_projects`(id, project_id)	Add a Service Report to a project
`put_services_shares_groups`(id, group_ids, ...)	Set the permissions groups has on this object
`put_services_shares_users`(id, user_ids, ...)	Set the permissions users have on this object
`put_services_transfer`(id, user_id, ...[, ...])	Transfer ownership of this object to another user
`put_shares_groups`(id, group_ids, ...[, ...])	Set the permissions groups has on this object
`put_shares_users`(id, user_ids, ...[, ...])	Set the permissions users have on this object
`put_transfer`(id, user_id, ...[, email_body, ...])	Transfer ownership of this object to another user

—

delete_grants(*id*)

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

> > **Parameters**

> > > **id** [integer] The ID of this report.

> > **Returns**

> > > **None** Response code 204: success

delete_projects(*id*, *project_id*)

> Remove a Report from a project

> > **Parameters**

> > > **id** [integer] The ID of the Report.

> > > **project_id** [integer] The ID of the project.

> > **Returns**

> > > **None** Response code 204: success

delete_services_projects(*id*, *project_id*)

> Remove a Service Report from a project

> > **Parameters**

> > > **id** [integer] The ID of the Service Report.

> > > **project_id** [integer] The ID of the project.

> > **Returns**

> > > **None** Response code 204: success

delete_services_shares_groups(*id*, *group_id*)

> Revoke the permissions a group has on this object

> > **Parameters**

> > > **id** [integer] The ID of the resource that is shared.

> > > **group_id** [integer] The ID of the group.

> > **Returns**

> > > **None** Response code 204: success

delete_services_shares_users(*id*, *user_id*)

> Revoke the permissions a user has on this object

> > **Parameters**

> > > **id** [integer] The ID of the resource that is shared.

> > > **user_id** [integer] The ID of the user.

> > **Returns**

> > > **None** Response code 204: success

delete_shares_groups(*id*, *group_id*)

> Revoke the permissions a group has on this object

> > **Parameters**

> > > **id** [integer] The ID of the resource that is shared.

> > > **group_id** [integer] The 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] The ID of the resource that is shared.

> > > **user_id** [integer] The 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

> [*civis.response.Response*](#)
>
> > - **id** [integer] The ID of this report.
> > - **name** [string] The name of the report.
> > - **user** [dict::]
> >
> > > - **id** [integer] The ID of this user.
> > > - **name** [string] This user's name.
> > > - **username** [string] This user's username.
> > > - **initials** [string] This user's initials.
> > > - **online** [boolean] Whether this user is online.
> >
> > - created_at : string/time
> > - updated_at : string/time
> > - **projects** [list::] A list of projects containing the report. - id : integer
> >
> > > The ID for the project.
> > >
> > > - **name** [string] The name of the project.
> >
> > - **state** [string] The status of the report's last run.
> > - **finished_at** [string/time] The time that the report's last run finished.
> > - **viz_updated_at** [string/time] The time that the report's visualization was last updated.
> > - **script** [dict::]
> >
> > > - **id** [integer] The ID for the script.
> > > - **name** [string] The name of the script.
> > > - **sql** [string] The raw SQL query for the script.
> >
> > - **job_path** [string] The link to details of the job that backs this report.
> > - tableau_id : integer
> > - type : string
> > - **template_id** [integer] The ID of the template used for this report.
> > - **auth_thumbnail_url** [string] URL for a thumbnail of the report.
> > - **last_run** [dict::]
> >
> > > - id : integer
> > > - state : string
> > > - **created_at** [string/time] The time that the run was queued.
> > > - **started_at** [string/time] The time that the run started.
> > > - **finished_at** [string/time] The time that the run completed.
> > > - **error** [string] The error message for this run, if present.
> >
> > - **archived** [string] The archival status of the requested item(s).
> > - **hidden** [boolean] The hidden status of the item.
> > - auth_data_url : string
> > - auth_code_url : string
> > - **config** [string] Any configuration metadata for this report.
> > - **valid_output_file** [boolean] Whether the job (a script or a query) that backs the report currently has a valid output file.

- **provide_api_key** [boolean] Whether the report requests an API Key from the report viewer.
- **api_key** [string] A Civis API key that can be used by this report.
- **api_key_id** [integer] The ID of the API key. Can be used for auditing API use by this report.
- **app_state** [dict] Any application state blob for this report.
- **use_viewers_tableau_username** [boolean] Apply user level filtering on Tableau reports.

get_git_commits(*id*, *commit_hash*)

   Get file contents at git ref

   Parameters

   id [integer] The ID of the file.

   commit_hash [string] The SHA (full or shortened) of the desired git commit.

   Returns

   civis.response.Response

   - **content** [string] The file's contents.
   - **type** [string] The file's type.
   - **size** [integer] The file's size.
   - **file_hash** [string] The SHA of the file.

get_services(*id*)

   Show a single service report

   Parameters

   id [integer] The ID of this report.

   Returns

   civis.response.Response

   - **id** [integer] The ID of this report.
   - **name** [string] The name of the report.
   - **user** [dict::]

        – **id** [integer] The ID of this user.

        – **name** [string] This user's name.

        – **username** [string] This user's username.

        – **initials** [string] This user's initials.

        – **online** [boolean] Whether this user is online.
   - created_at : string/time
   - updated_at : string/time
   - **host** [string] The host for the service report
   - **display_url** [string] The URL to display the service report.
   - **service_id** [integer] The id of the backing service
   - **provide_api_key** [boolean] Whether the report requests an API Key from the report viewer.
   - **api_key** [string] A Civis API key that can be used by this report.
   - **api_key_id** [integer] The ID of the API key. Can be used for auditing API use by this report.
   - **archived** [string] The archival status of the requested item(s).

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

   List Reports

   Parameters

> **type** [string, optional] If specified, return report of these types. It accepts a comma-separated list, possible values are 'tableau' or 'other'.
>
> **template_id** [integer, optional] If specified, return reports using the provided Template.
>
> **author** [string, optional] If specified, return items from any of these authors. It accepts a comma- separated list of user IDs.
>
> **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
>
> **archived** [string, optional] The archival status of the requested item(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

    civis.response.PaginatedResponse

- **id** [integer] The ID of this report.
- **name** [string] The name of the report.
- **user** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- created_at : string/time
- updated_at : string/time
- **projects** [list::] A list of projects containing the report. - id : integer

    The ID for the project.

    - **name** [string] The name of the project.

- **state** [string] The status of the report's last run.
- **finished_at** [string/time] The time that the report's last run finished.
- **viz_updated_at** [string/time] The time that the report's visualization was last updated.
- **script** [dict::]

    - **id** [integer] The ID for the script.

    - **name** [string] The name of the script.

    - **sql** [string] The raw SQL query for the script.

- **job_path** [string] The link to details of the job that backs this report.
- tableau_id : integer
- type : string
- **template_id** [integer] The ID of the template used for this report.
- **auth_thumbnail_url** [string] URL for a thumbnail of the report.

- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **archived** [string] The archival status of the requested item(s).

list_dependencies(*id*, ***, *user_id='DEFAULT'*)

   List dependent objects for this object

   Parameters

   id [integer] The ID of the resource that is shared.

   user_id [integer, optional] ID of target user

   Returns

   civis.response.Response

- **object_type** [string] Dependent object type
- **fco_type** [string] Human readable dependent object type
- **id** [integer] Dependent object ID
- **name** [string] Dependent object name, or nil if the requesting user cannot read this object
- **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
- **shareable** [boolean] Whether or not the requesting user can share this object.

list_git(*id*)

   Get the git metadata attached to an item

   Parameters

   id [integer] The ID of the file.

   Returns

   civis.response.Response

- **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
- **git_branch** [string] The git branch that the file is on.
- **git_path** [string] The path of the file in the repository.
- **git_repo** [dict::]

    - **id** [integer] The ID for this git repository.

    - **repo_url** [string] The URL for this git repository.

    - created_at : string/time

    - updated_at : string/time
- **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
- **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

list_git_commits(*id*)

   Get the git commits for an item on the current branch

   Parameters

> **id** [integer] The ID of the file.

Returns

civis.response.Response

- **commit_hash** [string] The SHA of the commit.
- **author_name** [string] The name of the commit's author.
- **date** [string/time] The commit's timestamp.
- **message** [string] The commit message.

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

List the projects a Report belongs to

Parameters

id [integer] The ID of the Report.

hidden [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.

Returns

civis.response.Response

- **id** [integer] The ID for this project.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- **name** [string] The name of this project.
- **description** [string] A description of the project.
- **users** [list::] Users who can see the project. - id : integer

    The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- auto_share : boolean
- created_at : string/time
- updated_at : string/time
- **archived** [string] The archival status of the requested item(s).

list_services_dependencies(*id*, ***, *user_id='DEFAULT'*)

List dependent objects for this object

Parameters

id [integer] The ID of the resource that is shared.

user_id [integer, optional] ID of target user

Returns

civis.response.Response

- **object_type** [string] Dependent object type
- **fco_type** [string] Human readable dependent object type
- **id** [integer] Dependent object ID
- **name** [string] Dependent object name, or nil if the requesting user cannot read this object

> • **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
>
> • **shareable** [boolean] Whether or not the requesting user can share this object.

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

> List the projects a Service Report belongs to
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the Service Report.
> > >
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > • **id** [integer] The ID for this project.
> > >
> > > • **author** [dict::]
> > >
> > > > – **id** [integer] The ID of this user.
> > > >
> > > > – **name** [string] This user's name.
> > > >
> > > > – **username** [string] This user's username.
> > > >
> > > > – **initials** [string] This user's initials.
> > > >
> > > > – **online** [boolean] Whether this user is online.
> > >
> > > • **name** [string] The name of this project.
> > >
> > > • **description** [string] A description of the project.
> > >
> > > • **users** [list::] Users who can see the project. - id : integer
> > >
> > > > The ID of this user.
> > > >
> > > > – **name** [string] This user's name.
> > > >
> > > > – **username** [string] This user's username.
> > > >
> > > > – **initials** [string] This user's initials.
> > > >
> > > > – **online** [boolean] Whether this user is online.
> > >
> > > • auto_share : boolean
> > >
> > > • created_at : string/time
> > >
> > > • updated_at : string/time
> > >
> > > • **archived** [string] The archival status of the requested item(s).

list_services_shares(*id*)

> List users and groups permissioned on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > • **readers** [dict::]
> > >
> > > > – **users** [list::]
> > > >
> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string
> > > >
> > > > – **groups** [list::]
> > > >
> > > > > ∗ id : integer
> > > > >
> > > > > ∗ name : string

- **writers** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **owners** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_shares(*id*)

　　List users and groups permissioned on this object

　　　　Parameters

　　　　　　id [integer] The ID of the resource that is shared.

　　　　Returns

　　　　　　civis.response.Response

- **readers** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **writers** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **owners** [dict::]

    - **users** [list::]

* id : integer

* name : string

– **groups** [list::]

* id : integer

* name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch(*id*, ***, *name='DEFAULT'*, *script_id='DEFAULT'*, *code_body='DEFAULT'*, *config='DEFAULT'*, *app_state='DEFAULT'*, *provide_api_key='DEFAULT'*, *template_id='DEFAULT'*, *use_viewers_tableau_username='DEFAULT'*)

Update a report

Parameters

id [integer] The ID of the report to modify.

name [string, optional] The name of the report.

script_id [integer, optional] The ID of the job (a script or a query) used to create this report.

code_body [string, optional] The code for the report visualization.

config [string, optional]

app_state [dict, optional] The application state blob for this report.

provide_api_key [boolean, optional] Allow the report to provide an API key to front-end code.

template_id [integer, optional] The ID of the template used for this report. If null is passed, no template will back this report. Changes to the backing template will reset the report appState.

use_viewers_tableau_username [boolean, optional] Apply user level filtering on Tableau reports.

Returns

civis.response.Response

- **id** [integer] The ID of this report.
- **name** [string] The name of the report.
- **user** [dict::]

– **id** [integer] The ID of this user.

– **name** [string] This user's name.

– **username** [string] This user's username.

– **initials** [string] This user's initials.

– **online** [boolean] Whether this user is online.

- created_at : string/time
- updated_at : string/time
- **projects** [list::] A list of projects containing the report. - id : integer

The ID for the project.

– **name** [string] The name of the project.

- **state** [string] The status of the report's last run.
- **finished_at** [string/time] The time that the report's last run finished.
- **viz_updated_at** [string/time] The time that the report's visualization was last updated.

- **script** [dict::]

    - **id** [integer] The ID for the script.

    - **name** [string] The name of the script.

    - **sql** [string] The raw SQL query for the script.
- **job_path** [string] The link to details of the job that backs this report.
- tableau_id : integer
- type : string
- **template_id** [integer] The ID of the template used for this report.
- **auth_thumbnail_url** [string] URL for a thumbnail of the report.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.
- auth_data_url : string
- auth_code_url : string
- **config** [string] Any configuration metadata for this report.
- **valid_output_file** [boolean] Whether the job (a script or a query) that backs the report currently has a valid output file.
- **provide_api_key** [boolean] Whether the report requests an API Key from the report viewer.
- **api_key** [string] A Civis API key that can be used by this report.
- **api_key_id** [integer] The ID of the API key. Can be used for auditing API use by this report.
- **app_state** [dict] Any application state blob for this report.
- **use_viewers_tableau_username** [boolean] Apply user level filtering on Tableau reports.

patch_git(*id*, ***, *git_ref='DEFAULT'*, *git_branch='DEFAULT'*, *git_path='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref_type='DEFAULT'*, *pull_from_git='DEFAULT'*)

Update an attached git file

Parameters

id [integer] The ID of the file.

git_ref [string, optional] A git reference specifying an unambiguous version of the file. Can be a branch name, or the full or shortened SHA of a commit.

git_branch [string, optional] The git branch that the file is on.

git_path [string, optional] The path of the file in the repository.

git_repo_url [string, optional] The URL of the git repository.

git_ref_type [string, optional] Specifies if the file is versioned by branch or tag.

pull_from_git [boolean, optional] Automatically pull latest commit from git. Only works for scripts.

Returns

civis.response.Response

- **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.

- **git_branch** [string] The git branch that the file is on.
- **git_path** [string] The path of the file in the repository.
- **git_repo** [dict::]

    – **id** [integer] The ID for this git repository.

    – **repo_url** [string] The URL for this git repository.

    – created_at : string/time

    – updated_at : string/time
- **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
- **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

patch_services(*id*, ***, *name='DEFAULT'*, *provide_api_key='DEFAULT'*)

> Update some attributes of this service report
>
> > **Parameters**
> >
> > > **id** [integer] The ID of this report.
> > >
> > > **name** [string, optional] The name of the service report.
> > >
> > > **provide_api_key** [boolean, optional] Whether the report requests an API Key from the report viewer.
> >
> > **Returns**
> >
> > [`civis.response.Response`](#)
> >
> > > - **id** [integer] The ID of this report.
> > > - **name** [string] The name of the report.
> > > - **user** [dict::]
> > >
> > >   – **id** [integer] The ID of this user.
> > >
> > >   – **name** [string] This user's name.
> > >
> > >   – **username** [string] This user's username.
> > >
> > >   – **initials** [string] This user's initials.
> > >
> > >   – **online** [boolean] Whether this user is online.
> > > - created_at : string/time
> > > - updated_at : string/time
> > > - **host** [string] The host for the service report
> > > - **display_url** [string] The URL to display the service report.
> > > - **service_id** [integer] The id of the backing service
> > > - **provide_api_key** [boolean] Whether the report requests an API Key from the report viewer.
> > > - **api_key** [string] A Civis API key that can be used by this report.
> > > - **api_key_id** [integer] The ID of the API key. Can be used for auditing API use by this report.
> > > - **archived** [string] The archival status of the requested item(s).

post(***, *script_id='DEFAULT'*, *name='DEFAULT'*, *code_body='DEFAULT'*, *app_state='DEFAULT'*, *provide_api_key='DEFAULT'*, *template_id='DEFAULT'*, *hidden='DEFAULT'*)

> Create a report
>
> > **Parameters**
> >
> > > **script_id** [integer, optional] The ID of the job (a script or a query) used to create this report.
> > >
> > > **name** [string, optional] The name of the report.
> > >
> > > **code_body** [string, optional] The code for the report visualization.
> > >
> > > **app_state** [dict, optional] Any application state blob for this report.

> **provide_api_key** [boolean, optional] Allow the report to provide an API key to front-
> end code.
>
> **template_id** [integer, optional] The ID of the template used for this report.
>
> **hidden** [boolean, optional] The hidden status of the item.

Returns

civis.response.Response

- **id** [integer] The ID of this report.
- **name** [string] The name of the report.
- **user** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- created_at : string/time
- updated_at : string/time
- **projects** [list::] A list of projects containing the report. - id : integer

        The ID for the project.

    - **name** [string] The name of the project.

- **state** [string] The status of the report's last run.
- **finished_at** [string/time] The time that the report's last run finished.
- **viz_updated_at** [string/time] The time that the report's visualization was
    last updated.
- **script** [dict::]

    - **id** [integer] The ID for the script.

    - **name** [string] The name of the script.

    - **sql** [string] The raw SQL query for the script.

- **job_path** [string] The link to details of the job that backs this report.
- tableau_id : integer
- type : string
- **template_id** [integer] The ID of the template used for this report.
- **auth_thumbnail_url** [string] URL for a thumbnail of the report.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.

- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.
- auth_data_url : string
- auth_code_url : string
- **config** [string] Any configuration metadata for this report.

- **valid_output_file** [boolean] Whether the job (a script or a query) that backs the report currently has a valid output file.
- **provide_api_key** [boolean] Whether the report requests an API Key from the report viewer.
- **api_key** [string] A Civis API key that can be used by this report.
- **api_key_id** [integer] The ID of the API key. Can be used for auditing API use by this report.
- **app_state** [dict] Any application state blob for this report.
- **use_viewers_tableau_username** [boolean] Apply user level filtering on Tableau reports.

post_git_checkout(*id*)

Checkout content that the existing git_ref points to and save to the object

> **Parameters**
>> **id** [integer] The ID of the file.
>
> **Returns**
>> *civis.response.Response*
>>> - **content** [string] The file's contents.
>>> - **type** [string] The file's type.
>>> - **size** [integer] The file's size.
>>> - **file_hash** [string] The SHA of the file.

post_git_checkout_latest(*id*)

Checkout latest commit on the current branch of a script or workflow

> **Parameters**
>> **id** [integer] The ID of the file.
>
> **Returns**
>> *civis.response.Response*
>>> - **content** [string] The file's contents.
>>> - **type** [string] The file's type.
>>> - **size** [integer] The file's size.
>>> - **file_hash** [string] The SHA of the file.

post_git_commits(*id*, *content*, *message*, *file_hash*)

Commit and push a new version of the file

> **Parameters**
>> **id** [integer] The ID of the file.
>>
>> **content** [string] The contents to commit to the file.
>>
>> **message** [string] A commit message describing the changes being made.
>>
>> **file_hash** [string] The full SHA of the file being replaced.
>
> **Returns**
>> *civis.response.Response*
>>> - **content** [string] The file's contents.
>>> - **type** [string] The file's type.
>>> - **size** [integer] The file's size.
>>> - **file_hash** [string] The SHA of the file.

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**
>> *civis.response.Response*
>>> - **id** [integer] The ID of this report.
>>> - **name** [string] The name of the report.
>>> - **user** [dict::]

> > > – **id** [integer] The ID of this user.
> >
> > > – **name** [string] This user's name.
> >
> > > – **username** [string] This user's username.
> >
> > > – **initials** [string] This user's initials.
> >
> > > – **online** [boolean] Whether this user is online.

- created_at : string/time
- updated_at : string/time
- **projects** [list::] A list of projects containing the report. - id : integer

    > The ID for the project.

    > > – **name** [string] The name of the project.

- **state** [string] The status of the report's last run.
- **finished_at** [string/time] The time that the report's last run finished.
- **viz_updated_at** [string/time] The time that the report's visualization was last updated.
- **script** [dict::]

    > > – **id** [integer] The ID for the script.
    > >
    > > – **name** [string] The name of the script.
    > >
    > > – **sql** [string] The raw SQL query for the script.

- **job_path** [string] The link to details of the job that backs this report.
- tableau_id : integer
- type : string
- **template_id** [integer] The ID of the template used for this report.
- **auth_thumbnail_url** [string] URL for a thumbnail of the report.
- **last_run** [dict::]

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

- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.
- auth_data_url : string
- auth_code_url : string
- **config** [string] Any configuration metadata for this report.
- **valid_output_file** [boolean] Whether the job (a script or a query) that backs the report currently has a valid output file.
- **provide_api_key** [boolean] Whether the report requests an API Key from the report viewer.
- **api_key** [string] A Civis API key that can be used by this report.
- **api_key_id** [integer] The ID of the API key. Can be used for auditing API use by this report.
- **app_state** [dict] Any application state blob for this report.
- **use_viewers_tableau_username** [boolean] Apply user level filtering on Tableau reports.

post_refresh(*id*)

> Refresh the data in this Tableau report

> > **Parameters**

> > > **id** [integer] The ID of this report.

> > **Returns**

> > > *civis.response.Response*

> > > > • **id** [integer] The ID of this report.

> > > > • **organization** [dict::]

> > > > > – **id** [integer] The ID of this organization.

> > > > > – **tableau_refresh_usage** [integer] The number of tableau re-freshes used this month.

> > > > > – **tableau_refresh_limit** [integer] The number of monthly tableau refreshes permitted to this organization.

> > > > > – **tableau_refresh_history** [list] The number of tableau re-freshes used this month.

post_services(*service_id*, ***, *provide_api_key='DEFAULT'*)

> Create a service report

> > **Parameters**

> > > **service_id** [integer] The id of the backing service

> > > **provide_api_key** [boolean, optional] Whether the report requests an API Key from the report viewer.

> > **Returns**

> > > *civis.response.Response*

> > > > • **id** [integer] The ID of this report.
> > > > • **name** [string] The name of the report.
> > > > • **user** [dict::]

> > > > > – **id** [integer] The ID of this user.

> > > > > – **name** [string] This user's name.

> > > > > – **username** [string] This user's username.

> > > > > – **initials** [string] This user's initials.

> > > > > – **online** [boolean] Whether this user is online.

> > > > • created_at : string/time
> > > > • updated_at : string/time
> > > > • **host** [string] The host for the service report
> > > > • **display_url** [string] The URL to display the service report.
> > > > • **service_id** [integer] The id of the backing service
> > > > • **provide_api_key** [boolean] Whether the report requests an API Key from the report viewer.
> > > > • **api_key** [string] A Civis API key that can be used by this report.
> > > > • **api_key_id** [integer] The ID of the API key. Can be used for auditing API use by this report.
> > > > • **archived** [string] The archival status of the requested item(s).

put_archive(*id*, *status*)

> Update the archive status of this object

> > **Parameters**

> > > **id** [integer] The ID of the object.
> > > **status** [boolean] The desired archived status of the object.

> > **Returns**

[`civis.response.Response`](civis.response.Response)
- **id** [integer] The ID of this report.
- **name** [string] The name of the report.
- **user** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- created_at : string/time
- updated_at : string/time
- **projects** [list::] A list of projects containing the report. - id : integer

    The ID for the project.

    - **name** [string] The name of the project.

- **state** [string] The status of the report's last run.
- **finished_at** [string/time] The time that the report's last run finished.
- **viz_updated_at** [string/time] The time that the report's visualization was
    last updated.
- **script** [dict::]

    - **id** [integer] The ID for the script.

    - **name** [string] The name of the script.

    - **sql** [string] The raw SQL query for the script.
- **job_path** [string] The link to details of the job that backs this report.
- tableau_id : integer
- type : string
- **template_id** [integer] The ID of the template used for this report.
- **auth_thumbnail_url** [string] URL for a thumbnail of the report.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.
- auth_data_url : string
- auth_code_url : string
- **config** [string] Any configuration metadata for this report.
- **valid_output_file** [boolean] Whether the job (a script or a query) that
    backs the report currently has a valid output file.
- **provide_api_key** [boolean] Whether the report requests an API Key from
    the report viewer.
- **api_key** [string] A Civis API key that can be used by this report.

- **api_key_id** [integer] The ID of the API key. Can be used for auditing API use by this report.
- **app_state** [dict] Any application state blob for this report.
- **use_viewers_tableau_username** [boolean] Apply user level filtering on Tableau reports.

put_git(*id*, ***, *git_ref='DEFAULT'*, *git_branch='DEFAULT'*, *git_path='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref_type='DEFAULT'*, *pull_from_git='DEFAULT'*)

Attach an item to a file in a git repo

Parameters

id [integer] The ID of the file.

git_ref [string, optional] A git reference specifying an unambiguous version of the file. Can be a branch name, or the full or shortened SHA of a commit.

git_branch [string, optional] The git branch that the file is on.

git_path [string, optional] The path of the file in the repository.

git_repo_url [string, optional] The URL of the git repository.

git_ref_type [string, optional] Specifies if the file is versioned by branch or tag.

pull_from_git [boolean, optional] Automatically pull latest commit from git. Only works for scripts.

Returns

civis.response.Response

- **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
- **git_branch** [string] The git branch that the file is on.
- **git_path** [string] The path of the file in the repository.
- **git_repo** [dict::]

  – **id** [integer] The ID for this git repository.

  – **repo_url** [string] The URL for this git repository.

  – created_at : string/time

  – updated_at : string/time

- **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
- **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

put_projects(*id*, *project_id*)

Add a Report to a project

Parameters

id [integer] The ID of the Report.

project_id [integer] The ID of the project.

Returns

None Response code 204: success

put_services_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

civis.response.Response

- **id** [integer] The ID of this report.
- **name** [string] The name of the report.

- **user** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- created_at : string/time
- updated_at : string/time
- **host** [string] The host for the service report
- **display_url** [string] The URL to display the service report.
- **service_id** [integer] The id of the backing service
- **provide_api_key** [boolean] Whether the report requests an API Key from the report viewer.
- **api_key** [string] A Civis API key that can be used by this report.
- **api_key_id** [integer] The ID of the API key. Can be used for auditing API use by this report.
- **archived** [string] The archival status of the requested item(s).

put_services_projects(*id*, *project_id*)

> Add a Service Report to a project
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the Service Report.
> > >
> > > **project_id** [integer] The ID of the project.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

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

> Set the permissions groups has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **group_ids** [list] An array of one or more group IDs.
> > >
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > >
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > >
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > - **readers** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > >   - **groups** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string
> > >
> > > - **writers** [dict::]
> > >
> > >   - **users** [list::]
> > >
> > >   * id : integer
> > >
> > >   * name : string

- – **groups** [list::]

    - ∗ id : integer

    - ∗ name : string

- • **owners** [dict::]

    - – **users** [list::]

        - ∗ id : integer

        - ∗ name : string

    - – **groups** [list::]

        - ∗ id : integer

        - ∗ name : string

- • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

- • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

Set the permissions users have on this object

Parameters

id [integer] The ID of the resource that is shared.

user_ids [list] An array of one or more user IDs.

permission_level [string] Options are: "read", "write", or "manage".

share_email_body [string, optional] Custom body text for e-mail sent on a share.

send_shared_email [boolean, optional] Send email to the recipients of a share.

Returns

[*civis.response.Response*](#)

- • **readers** [dict::]

    - – **users** [list::]

        - ∗ id : integer

        - ∗ name : string

    - – **groups** [list::]

        - ∗ id : integer

        - ∗ name : string

- • **writers** [dict::]

    - – **users** [list::]

        - ∗ id : integer

        - ∗ name : string

    - – **groups** [list::]

        - ∗ id : integer

        - ∗ name : string

- • **owners** [dict::]

    - – **users** [list::]

        - ∗ id : integer

           ∗ name : string

        – **groups** [list::]

           ∗ id : integer

           ∗ name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_services_transfer(*id*, *user_id*, *include_dependencies*, *, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

    Transfer ownership of this object to another user

        Parameters

           id [integer] The ID of the resource that is shared.

           user_id [integer] ID of target user

           include_dependencies [boolean] Whether or not to give manage permissions on all dependencies

           email_body [string, optional] Custom body text for e-mail sent on transfer.

           send_email [boolean, optional] Send email to the target user of the transfer?

        Returns

        civis.response.Response

- **dependencies** [list::] Dependent objects for this object - object_type : string

           Dependent object type

        – **fco_type** [string] Human readable dependent object type

        – **id** [integer] Dependent object ID

        – **name** [string] Dependent object name, or nil if the requesting user cannot read this object

        – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user

        – **shared** [boolean] Whether dependent object was successfully shared with target user

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

    Set the permissions groups has on this object

        Parameters

           id [integer] The ID of the resource that is shared.

           group_ids [list] An array of one or more group IDs.

           permission_level [string] Options are: "read", "write", or "manage".

           share_email_body [string, optional] Custom body text for e-mail sent on a share.

           send_shared_email [boolean, optional] Send email to the recipients of a share.

        Returns

        civis.response.Response

- **readers** [dict::]

        – **users** [list::]

           ∗ id : integer

       * name : string

     – **groups** [list::]

       * id : integer

       * name : string

   • **writers** [dict::]

     – **users** [list::]

       * id : integer

       * name : string

     – **groups** [list::]

       * id : integer

       * name : string

   • **owners** [dict::]

     – **users** [list::]

       * id : integer

       * name : string

     – **groups** [list::]

       * id : integer

       * name : string

   • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

   • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

 Set the permissions users have on this object

  Parameters

   id [integer] The ID of the resource that is shared.

   user_ids [list] An array of one or more user IDs.

   permission_level [string] Options are: "read", "write", or "manage".

   share_email_body [string, optional] Custom body text for e-mail sent on a share.

   send_shared_email [boolean, optional] Send email to the recipients of a share.

  Returns

   civis.response.Response

     • **readers** [dict::]

       – **users** [list::]

         * id : integer

         * name : string

       – **groups** [list::]

         * id : integer

         * name : string

     • **writers** [dict::]

       – **users** [list::]

> > > > > * id : integer
> > > > >
> > > > > * name : string
> > > >
> > > > – **groups** [list::]
> > > >
> > > > > * id : integer
> > > > >
> > > > > * name : string
> > >
> > > • **owners** [dict::]
> > >
> > > > – **users** [list::]
> > > >
> > > > > * id : integer
> > > > >
> > > > > * name : string
> > > >
> > > > – **groups** [list::]
> > > >
> > > > > * id : integer
> > > > >
> > > > > * name : string
> > >
> > > • **total_user_shares** [integer] For owners, the number of total users shared.
> > > For writers and readers, the number of visible users shared.
> > > • **total_group_shares** [integer] For owners, the number of total groups
> > > shared. For writers and readers, the number of visible groups shared.

put_transfer(*id*, *user_id*, *include_dependencies*, ***, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

> Transfer ownership of this object to another user
>
> **Parameters**
>
> > **id** [integer] The ID of the resource that is shared.
> > **user_id** [integer] ID of target user
> > **include_dependencies** [boolean] Whether or not to give manage permissions on all
> > dependencies
> > **email_body** [string, optional] Custom body text for e-mail sent on transfer.
> > **send_email** [boolean, optional] Send email to the target user of the transfer?
>
> **Returns**
>
> > *civis.response.Response*
> >
> > > • **dependencies** [list::] Dependent objects for this object - object_type :
> > > string
> > >
> > > > Dependent object type
> > > >
> > > > – **fco_type** [string] Human readable dependent object type
> > > >
> > > > – **id** [integer] Dependent object ID
> > > >
> > > > – **name** [string] Dependent object name, or nil if the requesting
> > > > user cannot read this object
> > > >
> > > > – **permission_level** [string] Permission level of target user (not
> > > > user's groups) for dependent object, or null if no target
> > > > user
> > > >
> > > > – **shared** [boolean] Whether dependent object was successfully
> > > > shared with target user

Roles

class Roles(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.roles.list(...)
```

Methods

list(*[, limit, page_num, order, order_dir, ...])	List Roles

———

list(**, limit='DEFAULT', page_num='DEFAULT', order='DEFAULT', order_dir='DEFAULT',*
    iterator='DEFAULT')
    List Roles
        Parameters
            limit [integer, optional] Number of results to return. Defaults to 50. Maximum allowed
                is 1000.
            page_num [integer, optional] Page number of the results to return. Defaults to the first
                page, 1.
            order [string, optional] The field on which to order the result set. Defaults to id. Must
                be one of: id.
            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
        civis.response.PaginatedResponse
                • **id** [integer] ID of the Role.
                • **name** [string] The name of the Role.
                • **slug** [string] The slug.
                • **description** [string] The description of the Role.

Saml_Service_Providers

class **Saml_Service_Providers**(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.saml_service_providers.list_shares(...)
```

Methods

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
list_shares(id)	List users and groups permissioned on this object
put_shares_groups(id, group_ids, ...[, ...])	Set the permissions groups has on this object
put_shares_users(id, user_ids, ...[, ...])	Set the permissions users have on this object

———

delete_shares_groups(*id*, *group_id*)

Revoke the permissions a group has on this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>> **group_id** [integer] The 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] The ID of the resource that is shared.
>> **user_id** [integer] The ID of the user.
> **Returns**
>> **None** Response code 204: success

list_shares(*id*)

List users and groups permissioned on this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
> **Returns**
>> *civis.response.Response*
>>> • **readers** [dict::]
>>>> – **users** [list::]
>>>>> ∗ id : integer
>>>>> ∗ name : string
>>>> – **groups** [list::]
>>>>> ∗ id : integer

                                                * name : string
                                    • **writers** [dict::]

                                            – **users** [list::]

                                                    * id : integer

                                                    * name : string

                                            – **groups** [list::]

                                                    * id : integer

                                                    * name : string

                                    • **owners** [dict::]

                                            – **users** [list::]

                                                    * id : integer

                                                    * name : string

                                            – **groups** [list::]

                                                    * id : integer

                                                    * name : string

                                    • **total_user_shares** [integer] For owners, the number of total users shared.
                                      For writers and readers, the number of visible users shared.
                                    • **total_group_shares** [integer] For owners, the number of total groups
                                      shared. For writers and readers, the number of visible groups shared.

put_shares_groups(*id*, *group_ids*, *permission_level*, *, *share_email_body='DEFAULT'*,
            send_shared_email='DEFAULT')
    Set the permissions groups has on this object
            Parameters
                    id [integer] The ID of the resource that is shared.
                    group_ids [list] An array of one or more group IDs.
                    permission_level [string] Options are: "read", "write", or "manage".
                    share_email_body [string, optional] Custom body text for e-mail sent on a share.
                    send_shared_email [boolean, optional] Send email to the recipients of a share.
            Returns
            [*civis.response.Response*](#)
                                    • **readers** [dict::]

                                            – **users** [list::]

                                                    * id : integer

                                                    * name : string

                                            – **groups** [list::]

                                                    * id : integer

                                                    * name : string

                                    • **writers** [dict::]

                                            – **users** [list::]

                                                    * id : integer

                                                    * name : string

                                            – **groups** [list::]

> > > > > > > > > ∗ id : integer
> > > > > > > > > ∗ name : string
> > > > > > > • **owners** [dict::]
> > > > > > > > – **users** [list::]
> > > > > > > > > ∗ id : integer
> > > > > > > > > ∗ name : string
> > > > > > > > – **groups** [list::]
> > > > > > > > > ∗ id : integer
> > > > > > > > > ∗ name : string
> > > > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object

> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_ids** [list] An array of one or more user IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.

> > **Returns**
> > [*civis.response.Response*](#)

> > > > • **readers** [dict::]
> > > > > – **users** [list::]
> > > > > > ∗ id : integer
> > > > > > ∗ name : string
> > > > > – **groups** [list::]
> > > > > > ∗ id : integer
> > > > > > ∗ name : string
> > > > • **writers** [dict::]
> > > > > – **users** [list::]
> > > > > > ∗ id : integer
> > > > > > ∗ name : string
> > > > > – **groups** [list::]
> > > > > > ∗ id : integer
> > > > > > ∗ name : string
> > > > • **owners** [dict::]
> > > > > – **users** [list::]
> > > > > > ∗ id : integer
> > > > > > ∗ name : string

>   – **groups** [list::]

>   ∗ id : integer

>   ∗ name : string
> - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

Scripts

class **Scripts**(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.scripts.list_types(...)
```

Methods

delete_containers_projects(id, project_id)	Remove a Container Script 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_projects(id, project_id)	Remove a Custom Script 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_projects(id, project_id)	Remove a JavaScript Script 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
delete_python3_projects(id, project_id)	Remove a Python Script 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_projects(id, project_id)	Remove an R Script 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_projects(id, project_id)	Remove a SQL script 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

continues on next page

Table 65 – continued from previous page

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 Custom Script
get_custom_runs(id, run_id)	Check status of a run
get_javascript(id)	Get a JavaScript Script
get_javascript_git_commits(id, commit_hash)	Get file contents at git ref
get_javascript_runs(id, run_id)	Check status of a run
get_python3(id)	Get a Python Script
get_python3_git_commits(id, commit_hash)	Get file contents at git ref
get_python3_runs(id, run_id)	Check status of a run
get_r(id)	Get an R Script
get_r_git_commits(id, commit_hash)	Get file contents at git ref
get_r_runs(id, run_id)	Check status of a run
get_sql(id)	Get a SQL script
get_sql_git_commits(id, commit_hash)	Get file contents at git ref
get_sql_runs(id, run_id)	Check status of a run
list(*[, type, category, author, status, ...])	List Scripts
list_containers_dependencies(id, *[, user_id])	List dependent objects for this object
list_containers_projects(id, *[, hidden])	List the projects a Container Script belongs to
list_containers_runs(id, *[, limit, ...])	List runs for the given container
list_containers_runs_logs(id, run_id, *[, ...])	Get the logs for a run
list_containers_runs_outputs(id, run_id, *)	List the outputs for a run
list_containers_shares(id)	List users and groups permissioned on this object
list_custom(*[, from_template_id, author, ...])	List Custom Scripts
list_custom_dependencies(id, *[, user_id])	List dependent objects for this object
list_custom_projects(id, *[, hidden])	List the projects a Custom Script belongs to
list_custom_runs(id, *[, limit, page_num, ...])	List runs for the given custom
list_custom_runs_logs(id, run_id, *[, ...])	Get the logs for a run
list_custom_runs_outputs(id, run_id, *[, ...])	List the outputs 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_dependencies(id, *[, user_id])	List dependent objects for this object
list_javascript_git(id)	Get the git metadata attached to an item
list_javascript_git_commits(id)	Get the git commits for an item on the current branch
list_javascript_projects(id, *[, hidden])	List the projects a JavaScript Script belongs to
list_javascript_runs(id, *[, limit, ...])	List runs for the given javascript
list_javascript_runs_logs(id, run_id, *[, ...])	Get the logs for a run
list_javascript_runs_outputs(id, run_id, *)	List the outputs for a run
list_javascript_shares(id)	List users and groups permissioned on this object
list_python3_dependencies(id, *[, user_id])	List dependent objects for this object
list_python3_git(id)	Get the git metadata attached to an item
list_python3_git_commits(id)	Get the git commits for an item on the current branch
list_python3_projects(id, *[, hidden])	List the projects a Python Script belongs to
list_python3_runs(id, *[, limit, page_num, ...])	List runs for the given python
list_python3_runs_logs(id, run_id, *[, ...])	Get the logs for a run
list_python3_runs_outputs(id, run_id, *[, ...])	List the outputs for a run
list_python3_shares(id)	List users and groups permissioned on this object

continues on next page

Table 65 – continued from previous page

list_r_dependencies(id, *[, user_id])	List dependent objects for this object
list_r_git(id)	Get the git metadata attached to an item
list_r_git_commits(id)	Get the git commits for an item on the current branch
list_r_projects(id, *[, hidden])	List the projects an R Script belongs to
list_r_runs(id, *[, limit, page_num, order, ...])	List runs for the given r
list_r_runs_logs(id, run_id, *[, last_id, limit])	Get the logs for a run
list_r_runs_outputs(id, run_id, *[, limit, ...])	List the outputs for a run
list_r_shares(id)	List users and groups permissioned on this object
list_sql_dependencies(id, *[, user_id])	List dependent objects for this object
list_sql_git(id)	Get the git metadata attached to an item
list_sql_git_commits(id)	Get the git commits for an item on the current branch
list_sql_projects(id, *[, hidden])	List the projects a SQL script belongs to
list_sql_runs(id, *[, limit, page_num, ...])	List runs for the given sql
list_sql_runs_logs(id, run_id, *[, last_id, ...])	Get the logs for a run
list_sql_runs_outputs(id, run_id, *[, ...])	List the outputs for a run
list_sql_shares(id)	List users and groups permissioned on this object
list_types()	List available script types
patch(id, *[, name, sql, params, arguments, ...])	Update a script
patch_container_runs(id, run_id, *[, error])	Update the given run
patch_containers(id, *[, name, parent_id, ...])	Update a container
patch_custom(id, *[, name, parent_id, ...])	Update some attributes of this Custom Script
patch_javascript(id, *[, name, parent_id, ...])	Update some attributes of this JavaScript Script
patch_javascript_git(id, *[, git_ref, ...])	Update an attached git file
patch_javascript_runs(id, run_id, *[, error])	Update the given run
patch_python3(id, *[, name, parent_id, ...])	Update some attributes of this Python Script
patch_python3_git(id, *[, git_ref, ...])	Update an attached git file
patch_python3_runs(id, run_id, *[, error])	Update the given run
patch_r(id, *[, name, parent_id, ...])	Update some attributes of this R Script
patch_r_git(id, *[, git_ref, git_branch, ...])	Update an attached git file
patch_r_runs(id, run_id, *[, error])	Update the given run
patch_sql(id, *[, name, parent_id, ...])	Update some attributes of this SQL script
patch_sql_git(id, *[, git_ref, git_branch, ...])	Update an attached git file
patch_sql_runs(id, run_id, *[, error])	Update the given run
post(name, remote_host_id, credential_id, sql, *)	Create a script
post_cancel(id)	Cancel a run
post_containers(required_resources, ...[, ...])	Create a container
post_containers_clone(id, *[, ...])	Clone this Container Script
post_containers_runs(id)	Start a run
post_containers_runs_logs(id, run_id, *[, ...])	Add log messages
post_containers_runs_outputs(id, run_id, ...)	Add an output for a run
post_custom(from_template_id, *[, name, ...])	Create a Custom Script
post_custom_clone(id, *[, clone_schedule, ...])	Clone this Custom Script
post_custom_runs(id)	Start a run
post_custom_runs_outputs(id, run_id, ...)	Add an output for a run
post_javascript(name, source, ...[, ...])	Create a JavaScript Script
post_javascript_clone(id, *[, ...])	Clone this JavaScript Script
post_javascript_git_checkout(id)	Checkout content that the existing git_ref points to and save to the object
post_javascript_git_checkout_latest(id)	Checkout latest commit on the current branch of a script or workflow

continues on next page

Table 65 – continued from previous page

post_javascript_git_commits(id, content, ...)	Commit and push a new version of the file
post_javascript_runs(id)	Start a run
post_javascript_runs_outputs(id, run_id, ...)	Add an output for a run
post_python3(name, source, *[, parent_id, ...])	Create a Python Script
post_python3_clone(id, *[, clone_schedule, ...])	Clone this Python Script
post_python3_git_checkout(id)	Checkout content that the existing git_ref points to and save to the object
post_python3_git_checkout_latest(id)	Checkout latest commit on the current branch of a script or workflow
post_python3_git_commits(id, content, ...)	Commit and push a new version of the file
post_python3_runs(id)	Start a run
post_python3_runs_outputs(id, run_id, ...)	Add an output for a run
post_r(name, source, *[, parent_id, ...])	Create an R Script
post_r_clone(id, *[, clone_schedule, ...])	Clone this R Script
post_r_git_checkout(id)	Checkout content that the existing git_ref points to and save to the object
post_r_git_checkout_latest(id)	Checkout latest commit on the current branch of a script or workflow
post_r_git_commits(id, content, message, ...)	Commit and push a new version of the file
post_r_runs(id)	Start a run
post_r_runs_outputs(id, run_id, object_type, ...)	Add an output for a run
post_run(id)	Run a script
post_sql(name, sql, remote_host_id, ...[, ...])	Create a SQL script
post_sql_clone(id, *[, clone_schedule, ...])	Clone this SQL script
post_sql_git_checkout(id)	Checkout content that the existing git_ref points to and save to the object
post_sql_git_checkout_latest(id)	Checkout latest commit on the current branch of a script or workflow
post_sql_git_commits(id, content, message, ...)	Commit and push a new version of the file
post_sql_runs(id)	Start a run
put_containers(id, required_resources, ...)	Edit a container
put_containers_archive(id, status)	Update the archive status of this object
put_containers_projects(id, project_id)	Add a Container Script 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_containers_transfer(id, user_id, ...[, ...])	Transfer ownership of this object to another user
put_custom(id, *[, name, parent_id, ...])	Replace all attributes of this Custom Script
put_custom_archive(id, status)	Update the archive status of this object
put_custom_projects(id, project_id)	Add a Custom Script 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_custom_transfer(id, user_id, ...[, ...])	Transfer ownership of this object to another user
put_javascript(id, name, source, ...[, ...])	Replace all attributes of this JavaScript Script
put_javascript_archive(id, status)	Update the archive status of this object
put_javascript_git(id, *[, git_ref, ...])	Attach an item to a file in a git repo
put_javascript_projects(id, project_id)	Add a JavaScript Script 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
put_javascript_transfer(id, user_id, ...[, ...])	Transfer ownership of this object to another user

continues on next page

Table 65 – continued from previous page

put_python3(id, name, source, *[, ...])	Replace all attributes of this Python Script
put_python3_archive(id, status)	Update the archive status of this object
put_python3_git(id, *[, git_ref, ...])	Attach an item to a file in a git repo
put_python3_projects(id, project_id)	Add a Python Script 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_python3_transfer(id, user_id, ...[, ...])	Transfer ownership of this object to another user
put_r(id, name, source, *[, parent_id, ...])	Replace all attributes of this R Script
put_r_archive(id, status)	Update the archive status of this object
put_r_git(id, *[, git_ref, git_branch, ...])	Attach an item to a file in a git repo
put_r_projects(id, project_id)	Add an R Script 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_r_transfer(id, user_id, ...[, ...])	Transfer ownership of this object to another user
put_sql(id, name, sql, remote_host_id, ...)	Replace all attributes of this SQL script
put_sql_archive(id, status)	Update the archive status of this object
put_sql_git(id, *[, git_ref, git_branch, ...])	Attach an item to a file in a git repo
put_sql_projects(id, project_id)	Add a SQL script 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
put_sql_transfer(id, user_id, ...[, ...])	Transfer ownership of this object to another user

—

delete_containers_projects(*id*, *project_id*)

Remove a Container Script from a project

> **Parameters**
>> **id** [integer] The ID of the Container Script.
>> **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] The ID of the resource that is shared.
>> **group_id** [integer] The 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] The ID of the resource that is shared.
>> **user_id** [integer] The ID of the user.
> **Returns**

> > **None** Response code 204: success

delete_custom_projects(*id*, *project_id*)
> Remove a Custom Script from a project
> > **Parameters**
> > > **id** [integer] The ID of the Custom Script.
> > > **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] The ID of the resource that is shared.
> > > **group_id** [integer] The 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] The ID of the resource that is shared.
> > > **user_id** [integer] The ID of the user.
> > **Returns**
> > > **None** Response code 204: success

delete_javascript_projects(*id*, *project_id*)
> Remove a JavaScript Script from a project
> > **Parameters**
> > > **id** [integer] The ID of the JavaScript Script.
> > > **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] The ID of the resource that is shared.
> > > **group_id** [integer] The 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] The ID of the resource that is shared.

            user_id [integer] The ID of the user.

        Returns

            None Response code 204: success

delete_python3_projects(*id*, *project_id*)

    Remove a Python Script from a project

        Parameters

            id [integer] The ID of the Python Script.

            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] The ID of the resource that is shared.

            group_id [integer] The 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] The ID of the resource that is shared.

            user_id [integer] The ID of the user.

        Returns

            None Response code 204: success

delete_r_projects(*id*, *project_id*)

    Remove an R Script from a project

        Parameters

            id [integer] The ID of the R Script.

            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] The ID of the resource that is shared.
> > >
> > > **group_id** [integer] The 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] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] The ID of the user.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

delete_sql_projects(*id*, *project_id*)

> Remove a SQL script from a project
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the SQL script.
> > >
> > > **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] The ID of the resource that is shared.
> > >
> > > **group_id** [integer] The 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] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] The 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**
> >
> > > *civis.response.Response*
> > >
> > > > • **id** [integer] The ID for the script.
> > > >
> > > > • **name** [string] The name of the script.
> > > >
> > > > • **type** [string] The type of script.

- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time this script was last updated.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    – **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    – **label** [string] The label to present to users when asking them for the value.

    – **description** [string] A short sentence or fragment describing this parameter to the end user.

    – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    – **required** [boolean] Whether this param is required.

    – **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    – **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    – **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.

- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    - **details** [string] The details link to get more information about the script.

    - **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **sql** [string] The raw SQL query for the script.
- **expanded_arguments** [dict] Expanded arguments for use in injecting into different environments.
- **template_script_id** [integer] The ID of the template script, if any.

get_containers(*id*)

    View a container

        Parameters

            id [integer] The ID for the script.

        Returns

            civis.response.Response

- **id** [integer] The ID for the script.
- **name** [string] The name of the container.
- **type** [string] The type of the script (e.g Container)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    - **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    – **label** [string] The label to present to users when asking them for the value.

    – **description** [string] A short sentence or fragment describing this parameter to the end user.

    – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    – **required** [boolean] Whether this param is required.

    – **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    – **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    – **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template script.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    – **details** [string] The details link to get more information about the script.

    – **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

    – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    – **scheduled_hours** [list] Hours of the day it is scheduled on.

> > – **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> >
> > – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
> >
> > – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
>
> • **notifications** [dict::]
>
> > – **urls** [list] URLs to receive a POST request at job completion
> >
> > – **success_email_subject** [string] Custom subject line for success e-mail.
> >
> > – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.
> >
> > – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.
> >
> > – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."
> >
> > – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.
> >
> > – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.
> >
> > – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.
> >
> > – **success_on** [boolean] If success email notifications are on.
> >
> > – **failure_on** [boolean] If failure email notifications are on.
>
> • **running_as** [dict::]
>
> > – **id** [integer] The ID of this user.
> >
> > – **name** [string] This user's name.
> >
> > – **username** [string] This user's username.
> >
> > – **initials** [string] This user's initials.
> >
> > – **online** [boolean] Whether this user is online.
>
> • **required_resources** [dict::]
>
> > – **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.
> >
> > – **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.
> >
> > – **disk_space** [number/float] The amount of disk space, in GB, to allocate for the container. This space will be used to hold the git repo configured for the container and anything your container writes to /tmp or /data. Fractional values (e.g. 0.25) are supported.
>
> • **repo_http_uri** [string] The location of a github repo to clone into the container, e.g. github.com/my-user/my-repo.git.

- **repo_ref** [string] The tag or branch of the github repo to clone into the container.
- **remote_host_credential_id** [integer] The id of the database credentials to pass into the environment of the container.
- **git_credential_id** [integer] The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.
- **docker_command** [string] The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]. Defaults to the Docker image's ENTRYPOINT/CMD.
- **docker_image_name** [string] The name of the docker image to pull from DockerHub.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.
- **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.
- **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **time_zone** [string] The time zone of this script.
- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **target_project_id** [integer] Target project to which script outputs will be added.
- **running_as_id** [integer] The ID of the runner of 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**
> >
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID of the run.
> > > - **container_id** [integer] The ID of the container.
> > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

- **is_cancel_requested** [boolean] True if run cancel requested, else false.
- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.
- **error** [string] The error, if any, returned by the run.
- **max_memory_usage** [number/float] If the run has finished, the maximum amount of memory used during the run, in MB.
- **max_cpu_usage** [number/float] If the run has finished, the maximum amount of cpu used during the run, in millicores.

get_custom(*id*)

    Get a Custom Script

        Parameters

            id [integer]

        Returns

        civis.response.Response

- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g Custom)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- category : string
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    – **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    – **label** [string] The label to present to users when asking them for the value.

    – **description** [string] A short sentence or fragment describing this parameter to the end user.

    – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    – **required** [boolean] Whether this param is required.

- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template script.
- **ui_report_url** [integer] The url of the custom HTML.
- **ui_report_id** [integer] The id of the report with the custom HTML.
- **ui_report_provide_api_key** [boolean] Whether the ui report requests an API Key from the report viewer.
- **template_script_name** [string] The name of the template script.
- **template_note** [string] The template's note.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **code_preview** [string] The code that this script will run with arguments inserted.
- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

- **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

- **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

- **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **time_zone** [string] The time zone of this script.

- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

- **hidden** [boolean] The hidden status of the item.

- **archived** [string] The archival status of the requested item(s).

- **target_project_id** [integer] Target project to which script outputs will be added.

- **last_successful_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

- **required_resources** [dict::]

  - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares.

> - **memory** [integer] The amount of RAM to allocate for the container (in MB).
>
> - **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.
>
> - **partition_label** [string] The partition label used to run this object. Only applicable for jobs using Docker.Not generally available. Beware this attribute may be removed in the future.
> - **running_as_id** [integer] The ID of the runner of this script.

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**
>
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID of the run.
> > > - **custom_id** [integer] The ID of the custom.
> > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > - **started_at** [string/time] The time the last run started at.
> > > - **finished_at** [string/time] The time the last run completed.
> > > - **error** [string] The error, if any, returned by the run.
> > > - **max_memory_usage** [number/float] If the run has finished, the maximum amount of memory used during the run, in MB. Only available if the backing script is a Python, R, or container script.
> > > - **max_cpu_usage** [number/float] If the run has finished, the maximum amount of cpu used during the run, in millicores. Only available if the backing script is a Python, R, or container script.

get_javascript(*id*)

> Get a JavaScript Script
>
> **Parameters**
>
> > **id** [integer]
>
> **Returns**
>
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID for the script.
> > > - **name** [string] The name of the script.
> > > - **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
> > > - **created_at** [string/time] The time this script was created.
> > > - **updated_at** [string/time] The time the script was last updated.
> > > - **author** [dict::]
> > >
> > > > - **id** [integer] The ID of this user.
> > > >
> > > > - **name** [string] This user's name.
> > > >
> > > > - **username** [string] This user's username.
> > > >
> > > > - **initials** [string] This user's initials.
> > > >
> > > > - **online** [boolean] Whether this user is online.

- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

  The ID for the project.

  - **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

  The variable's name as used within your code.

  - **label** [string] The label to present to users when asking them for the value.

  - **description** [string] A short sentence or fragment describing this parameter to the end user.

  - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

  - **required** [boolean] Whether this param is required.

  - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

  - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

  - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

  - **details** [string] The details link to get more information about the script.

  - **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

– **started_at** [string/time] The time that the run started.

– **finished_at** [string/time] The time that the run completed.

– **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **source** [string] The body/text of the script.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **running_as_id** [integer] The ID of the runner of this script.

get_javascript_git_commits(*id*, *commit_hash*)

Get file contents at git ref

    Parameters

        id [integer] The ID of the file.

        commit_hash [string] The SHA (full or shortened) of the desired git commit.

    Returns

    civis.response.Response

- **content** [string] The file's contents.
- **type** [string] The file's type.
- **size** [integer] The file's size.
- **file_hash** [string] The SHA of the file.

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

    civis.response.Response

- **id** [integer] The ID of the run.
- **javascript_id** [integer] The ID of the javascript.
- **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **is_cancel_requested** [boolean] True if run cancel requested, else false.
- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.
- **error** [string] The error, if any, returned by the run.

get_python3(*id*)

Get a Python Script

    Parameters

        id [integer]

    Returns

    civis.response.Response

- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
- **created_at** [string/time] The time this script was created.

- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    - **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.

- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    – **details** [string] The details link to get more information about the script.

    – **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

    – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    – **scheduled_hours** [list] Hours of the day it is scheduled on.

    – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    – **urls** [list] URLs to receive a POST request at job completion

    – **success_email_subject** [string] Custom subject line for success e-mail.

    – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    – **success_on** [boolean] If success email notifications are on.

    – **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **required_resources** [dict::]

    - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

    - **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

    - **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.
- **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.
- **source** [string] The body/text of the script.
- **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.
- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
- **running_as_id** [integer] The ID of the runner of this script.

get_python3_git_commits(*id*, *commit_hash*)

    Get file contents at git ref

        Parameters

            id [integer] The ID of the file.

            commit_hash [string] The SHA (full or shortened) of the desired git commit.

        Returns

            civis.response.Response

- **content** [string] The file's contents.
- **type** [string] The file's type.
- **size** [integer] The file's size.
- **file_hash** [string] The SHA of the file.

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

        civis.response.Response

- **id** [integer] The ID of the run.
- **python_id** [integer] The ID of the python.
- **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **is_cancel_requested** [boolean] True if run cancel requested, else false.
- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.
- **error** [string] The error, if any, returned by the run.
- **max_memory_usage** [number/float] If the run has finished, the maximum amount of memory used during the run, in MB.
- **max_cpu_usage** [number/float] If the run has finished, the maximum amount of cpu used during the run, in millicores.

get_r(*id*)

    Get an R Script

        Parameters

            id [integer]

        Returns

        civis.response.Response

- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

        The ID for the project.

    – **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script

- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    - **details** [string] The details link to get more information about the script.

    - **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

- **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

  - **urls** [list] URLs to receive a POST request at job completion

  - **success_email_subject** [string] Custom subject line for success e-mail.

  - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

  - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

  - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

  - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

  - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

  - **success_on** [boolean] If success email notifications are on.

  - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.

- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **required_resources** [dict::]

  – **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

  – **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

  – **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.

- **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.
- **source** [string] The body/text of the script.
- **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.
- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
- **running_as_id** [integer] The ID of the runner of this script.

get_r_git_commits(*id*, *commit_hash*)

Get file contents at git ref

    Parameters

        id [integer] The ID of the file.

        commit_hash [string] The SHA (full or shortened) of the desired git commit.

    Returns

        civis.response.Response

- **content** [string] The file's contents.
- **type** [string] The file's type.
- **size** [integer] The file's size.
- **file_hash** [string] The SHA of the file.

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

        civis.response.Response

- **id** [integer] The ID of the run.
- **r_id** [integer] The ID of the r.
- **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **is_cancel_requested** [boolean] True if run cancel requested, else false.
- **started_at** [string/time] The time the last run started at.

5.5. API Client

- **finished_at** [string/time] The time the last run completed.
- **error** [string] The error, if any, returned by the run.
- **max_memory_usage** [number/float] If the run has finished, the maximum amount of memory used during the run, in MB.
- **max_cpu_usage** [number/float] If the run has finished, the maximum amount of cpu used during the run, in millicores.

get_sql(*id*)

> Get a SQL script
>
> > **Parameters**
> >
> > > **id** [integer]
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > - **id** [integer] The ID for the script.
> > > - **name** [string] The name of the script.
> > > - **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
> > > - **created_at** [string/time] The time this script was created.
> > > - **updated_at** [string/time] The time the script was last updated.
> > > - **author** [dict::]
> > >
> > > > – **id** [integer] The ID of this user.
> > > >
> > > > – **name** [string] This user's name.
> > > >
> > > > – **username** [string] This user's username.
> > > >
> > > > – **initials** [string] This user's initials.
> > > >
> > > > – **online** [boolean] Whether this user is online.
> > >
> > > - **state** [string] The status of the script's last run.
> > > - **finished_at** [string/time] The time that the script's last run finished.
> > > - **category** [string] The category of the script.
> > > - **projects** [list::] A list of projects containing the script. - id : integer
> > >
> > > > The ID for the project.
> > >
> > > > – **name** [string] The name of the project.
> > >
> > > - **parent_id** [integer] The ID of the parent job that will trigger this script
> > > - **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
> > > - **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string
> > >
> > > > The variable's name as used within your code.
> > >
> > > > – **label** [string] The label to present to users when asking them for the value.
> > > >
> > > > – **description** [string] A short sentence or fragment describing this parameter to the end user.
> > > >
> > > > – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
> > > >
> > > > – **required** [boolean] Whether this param is required.

- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    - **details** [string] The details link to get more information about the script.

    - **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

- **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

- **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **sql** [string] The raw SQL query for the script.
- **expanded_arguments** [dict] Expanded arguments for use in injecting into different environments.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **code_preview** [string] The code that this script will run with arguments inserted.
- **csv_settings** [dict::]

  - **include_header** [boolean] Whether or not to include headers in the output data. Default: true

  - **compression** [string] The type of compression to use, if any, one of "none", "zip", or "gzip". Default: gzip

– **column_delimiter** [string] Which delimiter to use, one of "comma", "tab", or "pipe". Default: comma

– **unquoted** [boolean] Whether or not to quote fields. Default: false

– **force_multifile** [boolean] Whether or not the csv should be split into multiple files. Default: false

– **filename_prefix** [string] A user specified filename prefix for the output file to have. Default: null

– **max_file_size** [integer] The max file size, in MB, created files will be. Only available when force_multifile is true.

• **running_as_id** [integer] The ID of the runner of this script.

get_sql_git_commits(*id*, *commit_hash*)

Get file contents at git ref

> **Parameters**
>> **id** [integer] The ID of the file.
>> **commit_hash** [string] The SHA (full or shortened) of the desired git commit.
>
> **Returns**
> [*civis.response.Response*](#)
>> • **content** [string] The file's contents.
>> • **type** [string] The file's type.
>> • **size** [integer] The file's size.
>> • **file_hash** [string] The SHA of the file.

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**
> [*civis.response.Response*](#)
>> • **id** [integer] The ID of the run.
>> • **sql_id** [integer] The ID of the sql.
>> • **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
>> • **is_cancel_requested** [boolean] True if run cancel requested, else false.
>> • **started_at** [string/time] The time the last run started at.
>> • **finished_at** [string/time] The time the last run completed.
>> • **error** [string] The error, if any, returned by the run.
>> • **output** [list::] A list of the outputs of this script. - output_name : string
>>
>>> The name of the output file.
>>
>>> – **file_id** [integer] The unique ID of the output file.
>>
>>> – **path** [string] The temporary link to download this output file, valid for 36 hours.
>>
>> • **output_cached_on** [string/time] The time that the output was originally exported, if a cache entry was used by the run.

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

List Scripts

Parameters

type [string, optional] If specified, return items of these types. The valid types are sql, python3, javascript, r, and containers.

category [string, optional] A job category for filtering scripts. Must be one of script, import, export, and enhancement.

author [string, optional] If specified, return items from any of these authors. It accepts a comma- separated list of user IDs.

status [string, optional] If specified, returns items with one of these statuses. It accepts a comma- separated list, possible values are 'running', 'failed', 'succeeded', 'idle', 'scheduled'.

hidden [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.

archived [string, optional] The archival status of the requested item(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

civis.response.PaginatedResponse

- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    – **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **is_template** [boolean] Whether others scripts use this one as a template.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **links** [dict::]

    – **details** [string] The details link to get more information about the script.

> > > > > – **runs** [string] The runs link to get the run information list for
> > > > > this script.
> > > > • **time_zone** [string] The time zone of this script.
> > > > • **last_run** [dict::]

> > > > > – id : integer

> > > > > – state : string

> > > > > – **created_at** [string/time] The time that the run was queued.

> > > > > – **started_at** [string/time] The time that the run started.

> > > > > – **finished_at** [string/time] The time that the run completed.

> > > > > – **error** [string] The error message for this run, if present.
> > > > • **archived** [string] The archival status of the requested item(s).
> > > > • **template_script_id** [integer] The ID of the template script, if any.

list_containers_dependencies(*id*, ***, *user_id='DEFAULT'*)

> List dependent objects for this object

> > **Parameters**

> > > **id** [integer] The ID of the resource that is shared.
> > > **user_id** [integer, optional] ID of target user

> > **Returns**

> > *civis.response.Response*

> > > • **object_type** [string] Dependent object type
> > > • **fco_type** [string] Human readable dependent object type
> > > • **id** [integer] Dependent object ID
> > > • **name** [string] Dependent object name, or nil if the requesting user cannot
> > > read this object
> > > • **permission_level** [string] Permission level of target user (not user's
> > > groups) for dependent object, or null if no target user
> > > • **shareable** [boolean] Whether or not the requesting user can share this ob-
> > > ject.

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

> List the projects a Container Script belongs to

> > **Parameters**

> > > **id** [integer] The ID of the Container Script.
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to
> > > false, returning non-hidden items.

> > **Returns**

> > *civis.response.Response*

> > > • **id** [integer] The ID for this project.
> > > • **author** [dict::]

> > > > – **id** [integer] The ID of this user.

> > > > – **name** [string] This user's name.

> > > > – **username** [string] This user's username.

> > > > – **initials** [string] This user's initials.

> > > > – **online** [boolean] Whether this user is online.
> > > • **name** [string] The name of this project.
> > > • **description** [string] A description of the project.
> > > • **users** [list::] Users who can see the project. - id : integer

> > > > The ID of this user.

> > > – **name** [string] This user's name.
> > >
> > > – **username** [string] This user's username.
> > >
> > > – **initials** [string] This user's initials.
> > >
> > > – **online** [boolean] Whether this user is online.
> >
> > - auto_share : boolean
> > - created_at : string/time
> > - updated_at : string/time
> > - **archived** [string] The archival status of the requested item(s).

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

> List runs for the given container
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the container.
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed
> > > is 100.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first
> > > page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to id. Must
> > > be one of: id.
> > >
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc
> > > (descending) defaulting to desc.
> > >
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use
> > > when more results than the maximum allowed by limit are needed. When True,
> > > limit and page_num are ignored. Defaults to False.
> >
> > **Returns**
> >
> > > *civis.response.PaginatedResponse*
> > >
> > > > - **id** [integer] The ID of the run.
> > > > - **container_id** [integer] The ID of the container.
> > > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded'
> > > >   'failed' or 'cancelled'.
> > > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > > - **started_at** [string/time] The time the last run started at.
> > > > - **finished_at** [string/time] The time the last run completed.
> > > > - **error** [string] The error, if any, returned by the run.
> > > > - **max_memory_usage** [number/float] If the run has finished, the maxi-
> > > >   mum amount of memory used during the run, in MB.
> > > > - **max_cpu_usage** [number/float] If the run has finished, the maximum
> > > >   amount of cpu used during the run, in millicores.

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

> Get the logs for a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the container.
> > >
> > > **run_id** [integer] The ID of the run.
> > >
> > > **last_id** [integer, optional] The ID of the last log message received. Log entries with this
> > > ID value or lower will be omitted.Logs are sorted by ID if this value is provided,
> > > and are otherwise sorted by createdAt.
> > >
> > > **limit** [integer, optional] The maximum number of log messages to return. Default of
> > > 10000.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **id** [integer] The ID of the log.

- **created_at** [string/date-time] The time the log was created.
- **message** [string] The log message.
- **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

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

    List the outputs for a run

Parameters

    id [integer] The ID of the container script.

    run_id [integer] The ID of the run.

    limit [integer, optional] Number of results to return. Defaults to its maximum of 50.

    page_num [integer, optional] Page number of the results to return. Defaults to the first page, 1.

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

    order_dir [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

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

Returns

civis.response.PaginatedResponse

- **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
- **object_id** [integer] The ID of the output.
- **name** [string] The name of the output.
- **link** [string] The hypermedia link to the output.
- value : string

list_containers_shares(*id*)

    List users and groups permissioned on this object

Parameters

    id [integer] The ID of the resource that is shared.

Returns

civis.response.Response

- **readers** [dict::]
  - **users** [list::]
    - ∗ id : integer
    - ∗ name : string
  - **groups** [list::]
    - ∗ id : integer
    - ∗ name : string
- **writers** [dict::]
  - **users** [list::]
    - ∗ id : integer
    - ∗ name : string
  - **groups** [list::]
    - ∗ id : integer

> > > > > > ∗ name : string

> > > > • **owners** [dict::]

> > > > > > – **users** [list::]

> > > > > > > > > ∗ id : integer

> > > > > > > > > ∗ name : string

> > > > > > – **groups** [list::]

> > > > > > > > > ∗ id : integer

> > > > > > > > > ∗ name : string

> > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

> > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> List Custom Scripts

> > **Parameters**

> > > **from_template_id** [string, optional] If specified, return scripts based on the template with this ID. Specify multiple IDs as a comma-separated list.

> > > **author** [string, optional] If specified, return items from any of these authors. It accepts a comma- separated list of user IDs.

> > > **status** [string, optional] If specified, returns items with one of these statuses. It accepts a comma- separated list, possible values are 'running', 'failed', 'succeeded', 'idle', 'scheduled'.

> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.

> > > **archived** [string, optional] The archival status of the requested item(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**

> > > *civis.response.PaginatedResponse*

> > > > • **id** [integer] The ID for the script.
> > > > • **name** [string] The name of the script.
> > > > • **type** [string] The type of the script (e.g Custom)
> > > > • **created_at** [string/time] The time this script was created.
> > > > • **updated_at** [string/time] The time the script was last updated.
> > > > • **author** [dict::]

> > > > > > – **id** [integer] The ID of this user.

> > > > > > – **name** [string] This user's name.

> > > > > > – **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.

- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **projects** [list::] A list of projects containing the script. - id : integer

> The ID for the project.

- **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **from_template_id** [integer] The ID of the template script.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.
- **archived** [string] The archival status of the requested item(s).
- **last_successful_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

list_custom_dependencies(*id*, ***, *user_id='DEFAULT'*)

> List dependent objects for this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **user_id** [integer, optional] ID of target user
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > - **object_type** [string] Dependent object type
> > > - **fco_type** [string] Human readable dependent object type
> > > - **id** [integer] Dependent object ID
> > > - **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > > - **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > > - **shareable** [boolean] Whether or not the requesting user can share this object.

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

> List the projects a Custom Script belongs to
>
> > **Parameters**

>> **id** [integer] The ID of the Custom Script.
>> **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
>
> **Returns**
>> *civis.response.Response*
>>> - **id** [integer] The ID for this project.
>>> - **author** [dict::]
>>>
>>>> – **id** [integer] The ID of this user.
>>>>
>>>> – **name** [string] This user's name.
>>>>
>>>> – **username** [string] This user's username.
>>>>
>>>> – **initials** [string] This user's initials.
>>>>
>>>> – **online** [boolean] Whether this user is online.
>>>
>>> - **name** [string] The name of this project.
>>> - **description** [string] A description of the project.
>>> - **users** [list::] Users who can see the project. - id : integer
>>>
>>>> The ID of this user.
>>>
>>>> – **name** [string] This user's name.
>>>>
>>>> – **username** [string] This user's username.
>>>>
>>>> – **initials** [string] This user's initials.
>>>>
>>>> – **online** [boolean] Whether this user is online.
>>>
>>> - auto_share : boolean
>>> - created_at : string/time
>>> - updated_at : string/time
>>> - **archived** [string] The archival status of the requested item(s).

list_custom_runs(*id*, ***, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*,
>> *order_dir='DEFAULT'*, *iterator='DEFAULT'*)
>
> List runs for the given custom
>> **Parameters**
>>> **id** [integer] The ID of the custom.
>>> **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 100.
>>> **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
>>> **order** [string, optional] The field on which to order the result set. Defaults to id. Must be one of: id.
>>> **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
>>> **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
>>
>> **Returns**
>>> *civis.response.PaginatedResponse*
>>>> - **id** [integer] The ID of the run.
>>>> - **custom_id** [integer] The ID of the custom.
>>>> - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
>>>> - **is_cancel_requested** [boolean] True if run cancel requested, else false.

> - **started_at** [string/time] The time the last run started at.
> - **finished_at** [string/time] The time the last run completed.
> - **error** [string] The error, if any, returned by the run.
> - **max_memory_usage** [number/float] If the run has finished, the maximum amount of memory used during the run, in MB. Only available if the backing script is a Python, R, or container script.
> - **max_cpu_usage** [number/float] If the run has finished, the maximum amount of cpu used during the run, in millicores. Only available if the backing script is a Python, R, or container script.

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

Get the logs for a run

> **Parameters**
>> **id** [integer] The ID of the custom.
>>
>> **run_id** [integer] The ID of the run.
>>
>> **last_id** [integer, optional] The ID of the last log message received. Log entries with this ID value or lower will be omitted.Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.
>>
>> **limit** [integer, optional] The maximum number of log messages to return. Default of 10000.
>
> **Returns**
>> *civis.response.Response*
>>> - **id** [integer] The ID of the log.
>>> - **created_at** [string/date-time] The time the log was created.
>>> - **message** [string] The log message.
>>> - **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

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

List the outputs for a run

> **Parameters**
>> **id** [integer] The ID of the custom script.
>>
>> **run_id** [integer] The ID of the run.
>>
>> **limit** [integer, optional] Number of results to return. Defaults to its maximum of 50.
>>
>> **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
>>
>> **order** [string, optional] The field on which to order the result set. Defaults to created_at. Must be one of: created_at, id.
>>
>> **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
>>
>> **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
>
> **Returns**
>> *civis.response.PaginatedResponse*
>>> - **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
>>> - **object_id** [integer] The ID of the output.
>>> - **name** [string] The name of the output.
>>> - **link** [string] The hypermedia link to the output.
>>> - value : string

list_custom_shares(*id*)

List users and groups permissioned on this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>
> **Returns**
>> *civis.response.Response*
>>> • **readers** [dict::]
>>>
>>>> – **users** [list::]
>>>>
>>>>> ∗ id : integer
>>>>>
>>>>> ∗ name : string
>>>>
>>>> – **groups** [list::]
>>>>
>>>>> ∗ id : integer
>>>>>
>>>>> ∗ name : string
>>>
>>> • **writers** [dict::]
>>>
>>>> – **users** [list::]
>>>>
>>>>> ∗ id : integer
>>>>>
>>>>> ∗ name : string
>>>>
>>>> – **groups** [list::]
>>>>
>>>>> ∗ id : integer
>>>>>
>>>>> ∗ name : string
>>>
>>> • **owners** [dict::]
>>>
>>>> – **users** [list::]
>>>>
>>>>> ∗ id : integer
>>>>>
>>>>> ∗ name : string
>>>>
>>>> – **groups** [list::]
>>>>
>>>>> ∗ id : integer
>>>>>
>>>>> ∗ name : string
>>>
>>> • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
>>> • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_history(*id*)

> Get the run history and outputs of this script
>
> **Parameters**
>> **id** [integer] The ID for the script.
>
> **Returns**
>> *civis.response.Response*
>>> • **id** [integer] The ID of this run.
>>> • **sql_id** [integer] The ID of this sql.
>>> • **state** [string] The state of this run.
>>> • **is_cancel_requested** [boolean] True if run cancel requested, else false.
>>> • **finished_at** [string/time] The time that this run finished.
>>> • **error** [string] The error message for this run, if present.
>>> • **output** [list::] A list of the outputs of this script. - output_name : string
>>>
>>>> The name of the output file.

> – **file_id** [integer] The unique ID of the output file.
>
> – **path** [string] The temporary link to download this output file, valid for 36 hours.

list_javascript_dependencies(*id*, ***, *user_id='DEFAULT'*)

> List dependent objects for this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **user_id** [integer, optional] ID of target user
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **object_type** [string] Dependent object type
> > > > • **fco_type** [string] Human readable dependent object type
> > > > • **id** [integer] Dependent object ID
> > > > • **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > > > • **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > > > • **shareable** [boolean] Whether or not the requesting user can share this object.

list_javascript_git(*id*)

> Get the git metadata attached to an item
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the file.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
> > > > • **git_branch** [string] The git branch that the file is on.
> > > > • **git_path** [string] The path of the file in the repository.
> > > > • **git_repo** [dict::]
> > > >
> > > > > – **id** [integer] The ID for this git repository.
> > > > >
> > > > > – **repo_url** [string] The URL for this git repository.
> > > > >
> > > > > – created_at : string/time
> > > > >
> > > > > – updated_at : string/time
> > > >
> > > > • **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
> > > > • **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

list_javascript_git_commits(*id*)

> Get the git commits for an item on the current branch
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the file.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **commit_hash** [string] The SHA of the commit.
> > > > • **author_name** [string] The name of the commit's author.
> > > > • **date** [string/time] The commit's timestamp.
> > > > • **message** [string] The commit message.

---

list_javascript_projects(*id*, ***, *hidden='DEFAULT'*)
> List the projects a JavaScript Script belongs to
> > **Parameters**
> > > **id** [integer] The ID of the JavaScript Script.
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> > **Returns**
> > [*civis.response.Response*](#)
> > > - **id** [integer] The ID for this project.
> > > - **author** [dict::]
> > > > - **id** [integer] The ID of this user.
> > > > - **name** [string] This user's name.
> > > > - **username** [string] This user's username.
> > > > - **initials** [string] This user's initials.
> > > > - **online** [boolean] Whether this user is online.
> > > - **name** [string] The name of this project.
> > > - **description** [string] A description of the project.
> > > - **users** [list::] Users who can see the project. - id : integer
> > > > The ID of this user.
> > > > - **name** [string] This user's name.
> > > > - **username** [string] This user's username.
> > > > - **initials** [string] This user's initials.
> > > > - **online** [boolean] Whether this user is online.
> > > - auto_share : boolean
> > > - created_at : string/time
> > > - updated_at : string/time
> > > - **archived** [string] The archival status of the requested item(s).

list_javascript_runs(*id*, ***, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)
> List runs for the given javascript
> > **Parameters**
> > > **id** [integer] The ID of the javascript.
> > > **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 100.
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > > **order** [string, optional] The field on which to order the result set. Defaults to id. Must be one of: id.
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> > **Returns**
> > [*civis.response.PaginatedResponse*](#)
> > > - **id** [integer] The ID of the run.
> > > - **javascript_id** [integer] The ID of the javascript.

- **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **is_cancel_requested** [boolean] True if run cancel requested, else false.
- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.
- **error** [string] The error, if any, returned by the run.

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

Get the logs for a run

> **Parameters**
>> **id** [integer] The ID of the javascript.
>>
>> **run_id** [integer] The ID of the run.
>>
>> **last_id** [integer, optional] The ID of the last log message received. Log entries with this ID value or lower will be omitted.Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.
>>
>> **limit** [integer, optional] The maximum number of log messages to return. Default of 10000.
>
> **Returns**
>> *civis.response.Response*
>>> - **id** [integer] The ID of the log.
>>> - **created_at** [string/date-time] The time the log was created.
>>> - **message** [string] The log message.
>>> - **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

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

List the outputs for a run

> **Parameters**
>> **id** [integer] The ID of the javascript script.
>>
>> **run_id** [integer] The ID of the run.
>>
>> **limit** [integer, optional] Number of results to return. Defaults to its maximum of 50.
>>
>> **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
>>
>> **order** [string, optional] The field on which to order the result set. Defaults to created_at. Must be one of: created_at, id.
>>
>> **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
>>
>> **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
>
> **Returns**
>> *civis.response.PaginatedResponse*
>>> - **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
>>> - **object_id** [integer] The ID of the output.
>>> - **name** [string] The name of the output.
>>> - **link** [string] The hypermedia link to the output.
>>> - value : string

list_javascript_shares(*id*)

List users and groups permissioned on this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>
> **Returns**

> > > > > *civis.response.Response*
> > > > > > - **readers** [dict::]
> > > > > > > - **users** [list::]
> > > > > > > > * id : integer
> > > > > > > > * name : string
> > > > > > > - **groups** [list::]
> > > > > > > > * id : integer
> > > > > > > > * name : string
> > > > > > - **writers** [dict::]
> > > > > > > - **users** [list::]
> > > > > > > > * id : integer
> > > > > > > > * name : string
> > > > > > > - **groups** [list::]
> > > > > > > > * id : integer
> > > > > > > > * name : string
> > > > > > - **owners** [dict::]
> > > > > > > - **users** [list::]
> > > > > > > > * id : integer
> > > > > > > > * name : string
> > > > > > > - **groups** [list::]
> > > > > > > > * id : integer
> > > > > > > > * name : string
> > > > > > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > > > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_python3_dependencies(*id*, ***, *user_id='DEFAULT'*)

> List dependent objects for this object
>
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_id** [integer, optional] ID of target user
> > **Returns**
> > *civis.response.Response*
> > > - **object_type** [string] Dependent object type
> > > - **fco_type** [string] Human readable dependent object type
> > > - **id** [integer] Dependent object ID
> > > - **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > > - **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > > - **shareable** [boolean] Whether or not the requesting user can share this object.

list_python3_git(*id*)

> Get the git metadata attached to an item

> > > **Parameters**
> > > > **id** [integer] The ID of the file.
> > >
> > > **Returns**
> > > > *civis.response.Response*
> > > > > - **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
> > > > > - **git_branch** [string] The git branch that the file is on.
> > > > > - **git_path** [string] The path of the file in the repository.
> > > > > - **git_repo** [dict::]
> > > > >
> > > > > > - **id** [integer] The ID for this git repository.
> > > > > >
> > > > > > - **repo_url** [string] The URL for this git repository.
> > > > > >
> > > > > > - created_at : string/time
> > > > > >
> > > > > > - updated_at : string/time
> > > > >
> > > > > - **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
> > > > > - **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

list_python3_git_commits(*id*)
> Get the git commits for an item on the current branch
> > **Parameters**
> > > **id** [integer] The ID of the file.
> >
> > **Returns**
> > > *civis.response.Response*
> > > > - **commit_hash** [string] The SHA of the commit.
> > > > - **author_name** [string] The name of the commit's author.
> > > > - **date** [string/time] The commit's timestamp.
> > > > - **message** [string] The commit message.

list_python3_projects(*id*, ***, *hidden='DEFAULT'*)
> List the projects a Python Script belongs to
> > **Parameters**
> > > **id** [integer] The ID of the Python Script.
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> >
> > **Returns**
> > > *civis.response.Response*
> > > > - **id** [integer] The ID for this project.
> > > > - **author** [dict::]
> > > > >
> > > > > - **id** [integer] The ID of this user.
> > > > >
> > > > > - **name** [string] This user's name.
> > > > >
> > > > > - **username** [string] This user's username.
> > > > >
> > > > > - **initials** [string] This user's initials.
> > > > >
> > > > > - **online** [boolean] Whether this user is online.
> > > > - **name** [string] The name of this project.
> > > > - **description** [string] A description of the project.
> > > > - **users** [list::] Users who can see the project. - id : integer
> > > > >
> > > > > The ID of this user.

– **name** [string] This user's name.

– **username** [string] This user's username.

– **initials** [string] This user's initials.

– **online** [boolean] Whether this user is online.

- auto_share : boolean
- created_at : string/time
- updated_at : string/time
- **archived** [string] The archival status of the requested item(s).

list_python3_runs(*id*, *, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)

List runs for the given python

> **Parameters**
>> **id** [integer] The ID of the python.
>>
>> **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 100.
>>
>> **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
>>
>> **order** [string, optional] The field on which to order the result set. Defaults to id. Must be one of: id.
>>
>> **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
>>
>> **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
>
> **Returns**
>> [*civis.response.PaginatedResponse*](#)
>>> - **id** [integer] The ID of the run.
>>> - **python_id** [integer] The ID of the python.
>>> - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
>>> - **is_cancel_requested** [boolean] True if run cancel requested, else false.
>>> - **started_at** [string/time] The time the last run started at.
>>> - **finished_at** [string/time] The time the last run completed.
>>> - **error** [string] The error, if any, returned by the run.
>>> - **max_memory_usage** [number/float] If the run has finished, the maximum amount of memory used during the run, in MB.
>>> - **max_cpu_usage** [number/float] If the run has finished, the maximum amount of cpu used during the run, in millicores.

list_python3_runs_logs(*id*, *run_id*, *, *last_id='DEFAULT'*, *limit='DEFAULT'*)

Get the logs for a run

> **Parameters**
>> **id** [integer] The ID of the python.
>>
>> **run_id** [integer] The ID of the run.
>>
>> **last_id** [integer, optional] The ID of the last log message received. Log entries with this ID value or lower will be omitted.Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.
>>
>> **limit** [integer, optional] The maximum number of log messages to return. Default of 10000.
>
> **Returns**
>> [*civis.response.Response*](#)
>>> - **id** [integer] The ID of the log.

- **created_at** [string/date-time] The time the log was created.
- **message** [string] The log message.
- **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

list_python3_runs_outputs(*id*, *run_id*, ***, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)

> List the outputs for a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the python script.
> > >
> > > **run_id** [integer] The ID of the run.
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to its maximum of 50.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to created_at. Must be one of: created_at, id.
> > >
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > >
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> >
> > **Returns**
> >
> > > *civis.response.PaginatedResponse*
> > >
> > > > - **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
> > > > - **object_id** [integer] The ID of the output.
> > > > - **name** [string] The name of the output.
> > > > - **link** [string] The hypermedia link to the output.
> > > > - value : string

list_python3_shares(*id*)

> List users and groups permissioned on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **readers** [dict::]
> > > >
> > > >   - **users** [list::]
> > > >
> > > >   * id : integer
> > > >
> > > >   * name : string
> > > >
> > > >   - **groups** [list::]
> > > >
> > > >   * id : integer
> > > >
> > > >   * name : string
> > > >
> > > > - **writers** [dict::]
> > > >
> > > >   - **users** [list::]
> > > >
> > > >   * id : integer
> > > >
> > > >   * name : string
> > > >
> > > >   - **groups** [list::]
> > > >
> > > >   * id : integer

                                       ∗ name : string

- **owners** [dict::]

    - **users** [list::]

        ∗ id : integer

        ∗ name : string

    - **groups** [list::]

        ∗ id : integer

        ∗ name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_r_dependencies(*id*, ***, *user_id='DEFAULT'*)

    List dependent objects for this object

        Parameters

            id [integer] The ID of the resource that is shared.

            user_id [integer, optional] ID of target user

        Returns

            civis.response.Response

- **object_type** [string] Dependent object type
- **fco_type** [string] Human readable dependent object type
- **id** [integer] Dependent object ID
- **name** [string] Dependent object name, or nil if the requesting user cannot read this object
- **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
- **shareable** [boolean] Whether or not the requesting user can share this object.

list_r_git(*id*)

    Get the git metadata attached to an item

        Parameters

            id [integer] The ID of the file.

        Returns

            civis.response.Response

- **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
- **git_branch** [string] The git branch that the file is on.
- **git_path** [string] The path of the file in the repository.
- **git_repo** [dict::]

    - **id** [integer] The ID for this git repository.

    - **repo_url** [string] The URL for this git repository.

    - created_at : string/time

    - updated_at : string/time

- **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
- **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

list_r_git_commits(*id*)

> Get the git commits for an item on the current branch
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the file.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **commit_hash** [string] The SHA of the commit.
> > > > - **author_name** [string] The name of the commit's author.
> > > > - **date** [string/time] The commit's timestamp.
> > > > - **message** [string] The commit message.

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

> List the projects an R Script belongs to
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the R Script.
> > >
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **id** [integer] The ID for this project.
> > > > - **author** [dict::]
> > > >
> > > > > - **id** [integer] The ID of this user.
> > > > >
> > > > > - **name** [string] This user's name.
> > > > >
> > > > > - **username** [string] This user's username.
> > > > >
> > > > > - **initials** [string] This user's initials.
> > > > >
> > > > > - **online** [boolean] Whether this user is online.
> > > >
> > > > - **name** [string] The name of this project.
> > > > - **description** [string] A description of the project.
> > > > - **users** [list::] Users who can see the project. - id : integer
> > > >
> > > > > The ID of this user.
> > > > >
> > > > > - **name** [string] This user's name.
> > > > >
> > > > > - **username** [string] This user's username.
> > > > >
> > > > > - **initials** [string] This user's initials.
> > > > >
> > > > > - **online** [boolean] Whether this user is online.
> > > >
> > > > - auto_share : boolean
> > > > - created_at : string/time
> > > > - updated_at : string/time
> > > > - **archived** [string] The archival status of the requested item(s).

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

> List runs for the given r
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the r.
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 100.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.

> order [string, optional] The field on which to order the result set. Defaults to id. Must
> be one of: id.
>
> order_dir [string, optional] Direction in which to sort, either asc (ascending) or desc
> (descending) defaulting to desc.
>
> iterator [bool, optional] If True, return a generator to iterate over all responses. Use
> when more results than the maximum allowed by limit are needed. When True,
> limit and page_num are ignored. Defaults to False.
>
> **Returns**
>
> *civis.response.PaginatedResponse*
>
> - **id** [integer] The ID of the run.
> - **r_id** [integer] The ID of the r.
> - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded'
>   'failed' or 'cancelled'.
> - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> - **started_at** [string/time] The time the last run started at.
> - **finished_at** [string/time] The time the last run completed.
> - **error** [string] The error, if any, returned by the run.
> - **max_memory_usage** [number/float] If the run has finished, the maxi-
>   mum amount of memory used during the run, in MB.
> - **max_cpu_usage** [number/float] If the run has finished, the maximum
>   amount of cpu used during the run, in millicores.

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

> Get the logs for a run
>
> **Parameters**
>
> id [integer] The ID of the r.
>
> run_id [integer] The ID of the run.
>
> last_id [integer, optional] The ID of the last log message received. Log entries with this
> ID value or lower will be omitted.Logs are sorted by ID if this value is provided,
> and are otherwise sorted by createdAt.
>
> limit [integer, optional] The maximum number of log messages to return. Default of
> 10000.
>
> **Returns**
>
> *civis.response.Response*
>
> - **id** [integer] The ID of the log.
> - **created_at** [string/date-time] The time the log was created.
> - **message** [string] The log message.
> - **level** [string] The level of the log. One of un-
>   known,fatal,error,warn,info,debug.

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

> List the outputs for a run
>
> **Parameters**
>
> id [integer] The ID of the r script.
>
> run_id [integer] The ID of the run.
>
> limit [integer, optional] Number of results to return. Defaults to its maximum of 50.
>
> page_num [integer, optional] Page number of the results to return. Defaults to the first
> page, 1.
>
> order [string, optional] The field on which to order the result set. Defaults to cre-
> ated_at. Must be one of: created_at, id.
>
> order_dir [string, optional] Direction in which to sort, either asc (ascending) or desc
> (descending) defaulting to desc.
>
> iterator [bool, optional] If True, return a generator to iterate over all responses. Use
> when more results than the maximum allowed by limit are needed. When True,

limit and page_num are ignored. Defaults to False.

> **Returns**
>> [`civis.response.PaginatedResponse`](#)
>>> • **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
>>> • **object_id** [integer] The ID of the output.
>>> • **name** [string] The name of the output.
>>> • **link** [string] The hypermedia link to the output.
>>> • value : string

list_r_shares(*id*)

> List users and groups permissioned on this object
>
>> **Parameters**
>>> **id** [integer] The ID of the resource that is shared.
>>
>> **Returns**
>>> [`civis.response.Response`](#)
>>> • **readers** [dict::]
>>>> – **users** [list::]
>>>>> * id : integer
>>>>> * name : string
>>>> – **groups** [list::]
>>>>> * id : integer
>>>>> * name : string
>>> • **writers** [dict::]
>>>> – **users** [list::]
>>>>> * id : integer
>>>>> * name : string
>>>> – **groups** [list::]
>>>>> * id : integer
>>>>> * name : string
>>> • **owners** [dict::]
>>>> – **users** [list::]
>>>>> * id : integer
>>>>> * name : string
>>>> – **groups** [list::]
>>>>> * id : integer
>>>>> * name : string
>>> • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
>>> • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_sql_dependencies(*id*, ***, *user_id='DEFAULT'*)

> List dependent objects for this object
>
>> **Parameters**
>>> **id** [integer] The ID of the resource that is shared.

> > > > > **user_id** [integer, optional] ID of target user
> > >
> > > **Returns**
> > >
> > > *civis.response.Response*
> > >
> > > > > - **object_type** [string] Dependent object type
> > > > > - **fco_type** [string] Human readable dependent object type
> > > > > - **id** [integer] Dependent object ID
> > > > > - **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > > > > - **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > > > > - **shareable** [boolean] Whether or not the requesting user can share this object.

list_sql_git(*id*)

> Get the git metadata attached to an item
>
> > **Parameters**
> >
> > **id** [integer] The ID of the file.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > > - **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
> > > > - **git_branch** [string] The git branch that the file is on.
> > > > - **git_path** [string] The path of the file in the repository.
> > > > - **git_repo** [dict::]
> > > >
> > > > > > - **id** [integer] The ID for this git repository.
> > > > > >
> > > > > > - **repo_url** [string] The URL for this git repository.
> > > > > >
> > > > > > - created_at : string/time
> > > > > >
> > > > > > - updated_at : string/time
> > > >
> > > > - **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
> > > > - **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

list_sql_git_commits(*id*)

> Get the git commits for an item on the current branch
>
> > **Parameters**
> >
> > **id** [integer] The ID of the file.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > > - **commit_hash** [string] The SHA of the commit.
> > > > - **author_name** [string] The name of the commit's author.
> > > > - **date** [string/time] The commit's timestamp.
> > > > - **message** [string] The commit message.

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

> List the projects a SQL script belongs to
>
> > **Parameters**
> >
> > **id** [integer] The ID of the SQL script.
> >
> > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> >
> > **Returns**
> >
> > *civis.response.Response*

- **id** [integer] The ID for this project.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **name** [string] The name of this project.
- **description** [string] A description of the project.
- **users** [list::] Users who can see the project. - id : integer

    The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- auto_share : boolean
- created_at : string/time
- updated_at : string/time
- **archived** [string] The archival status of the requested item(s).

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

   List runs for the given sql

   Parameters

   id [integer] The ID of the sql.

   limit [integer, optional] Number of results to return. Defaults to 20. Maximum allowed
       is 100.

   page_num [integer, optional] Page number of the results to return. Defaults to the first
       page, 1.

   order [string, optional] The field on which to order the result set. Defaults to id. Must
       be one of: id.

   order_dir [string, optional] Direction in which to sort, either asc (ascending) or desc
       (descending) defaulting to desc.

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

   Returns

   civis.response.PaginatedResponse

   - **id** [integer] The ID of the run.
   - **sql_id** [integer] The ID of the sql.
   - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded'
         'failed' or 'cancelled'.
   - **is_cancel_requested** [boolean] True if run cancel requested, else false.
   - **started_at** [string/time] The time the last run started at.
   - **finished_at** [string/time] The time the last run completed.
   - **error** [string] The error, if any, returned by the run.
   - **output** [list::] A list of the outputs of this script. - output_name : string

       The name of the output file.

> > > > – **file_id** [integer] The unique ID of the output file.
> > > >
> > > > – **path** [string] The temporary link to download this output file, valid for 36 hours.
> > >
> > > • **output_cached_on** [string/time] The time that the output was originally exported, if a cache entry was used by the run.

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

> Get the logs for a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the sql.
> > >
> > > **run_id** [integer] The ID of the run.
> > >
> > > **last_id** [integer, optional] The ID of the last log message received. Log entries with this ID value or lower will be omitted.Logs are sorted by ID if this value is provided, and are otherwise sorted by createdAt.
> > >
> > > **limit** [integer, optional] The maximum number of log messages to return. Default of 10000.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **id** [integer] The ID of the log.
> > > > • **created_at** [string/date-time] The time the log was created.
> > > > • **message** [string] The log message.
> > > > • **level** [string] The level of the log. One of unknown,fatal,error,warn,info,debug.

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

> List the outputs for a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the sql script.
> > >
> > > **run_id** [integer] The ID of the run.
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to its maximum of 50.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to created_at. Must be one of: created_at, id.
> > >
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > >
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> >
> > **Returns**
> >
> > > [*civis.response.PaginatedResponse*](#)
> > >
> > > > • **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
> > > > • **object_id** [integer] The ID of the output.
> > > > • **name** [string] The name of the output.
> > > > • **link** [string] The hypermedia link to the output.
> > > > • value : string

list_sql_shares(*id*)

> List users and groups permissioned on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> >
> > **Returns**

> > > > *civis.response.Response*
> > > > > • **readers** [dict::]
> > > > > > – **users** [list::]
> > > > > > > * id : integer
> > > > > > >
> > > > > > > * name : string
> > > > > >
> > > > > > – **groups** [list::]
> > > > > > > * id : integer
> > > > > > >
> > > > > > > * name : string
> > > > >
> > > > > • **writers** [dict::]
> > > > > > – **users** [list::]
> > > > > > > * id : integer
> > > > > > >
> > > > > > > * name : string
> > > > > >
> > > > > > – **groups** [list::]
> > > > > > > * id : integer
> > > > > > >
> > > > > > > * name : string
> > > > >
> > > > > • **owners** [dict::]
> > > > > > – **users** [list::]
> > > > > > > * id : integer
> > > > > > >
> > > > > > > * name : string
> > > > > >
> > > > > > – **groups** [list::]
> > > > > > > * id : integer
> > > > > > >
> > > > > > > * name : string
> > > > >
> > > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_types()

> List available script types
> > **Returns**
> > > *civis.response.Response*
> > > > • **name** [string] The name of the type.

patch(*id*, ***, *name='DEFAULT'*, *sql='DEFAULT'*, *params='DEFAULT'*, *arguments='DEFAULT'*, *template_script_id='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*, *parent_id='DEFAULT'*, *running_as_id='DEFAULT'*)

> Update a script
> > **Parameters**
> > > **id** [integer] The ID for the script.
> > > **name** [string, optional] The name of the script.
> > > **sql** [string, optional] The raw SQL query for the script.
> > > **params** [list, optional::] A definition of the parameters this script accepts in the arguments field. Cannot be set if this script uses a template script. - name : string The variable's name as used within your code.
> > > > • **label** [string] The label to present to users when asking them for the value.

- **description** [string] A short sentence or fragment describing this parameter to the end user.
- **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
- **required** [boolean] Whether this param is required.
- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.
- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

arguments [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.

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

schedule [dict, optional::]
- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

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

parent_id [integer, optional] The ID of the parent job that will trigger this script

running_as_id [integer, optional] The ID of the runner of this script.

Returns

civis.response.Response
- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of script.

- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time this script was last updated.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    – **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    – **label** [string] The label to present to users when asking them for the value.

    – **description** [string] A short sentence or fragment describing this parameter to the end user.

    – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    – **required** [boolean] Whether this param is required.

    – **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    – **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    – **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.

- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    – **details** [string] The details link to get more information about the script.

    – **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

    – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    – **scheduled_hours** [list] Hours of the day it is scheduled on.

    – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    – **urls** [list] URLs to receive a POST request at job completion

    – **success_email_subject** [string] Custom subject line for success e-mail.

    – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    – **success_on** [boolean] If success email notifications are on.

    – **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **sql** [string] The raw SQL query for the script.
- **expanded_arguments** [dict] Expanded arguments for use in injecting into different environments.
- **template_script_id** [integer] The ID of the template script, if any.

patch_container_runs(*id*, *run_id*, *, error='DEFAULT'*)

> Update the given run
>> **Parameters**
>>> **id** [integer] ID of the Job
>>> **run_id** [integer] ID of the Run
>>> **error** [string, optional] The error message to update
>> **Returns**
>>> **None** Response code 204: success

patch_containers(*id*, *, name='DEFAULT'*, *parent_id='DEFAULT'*, *user_context='DEFAULT'*,
               params='DEFAULT', *arguments='DEFAULT'*, *schedule='DEFAULT'*,
               notifications='DEFAULT', *required_resources='DEFAULT'*, *repo_http_uri='DEFAULT'*,
               repo_ref='DEFAULT', *remote_host_credential_id='DEFAULT'*,
               git_credential_id='DEFAULT', *docker_command='DEFAULT'*,
               docker_image_name='DEFAULT', *docker_image_tag='DEFAULT'*,
               instance_type='DEFAULT', *cancel_timeout='DEFAULT'*, *time_zone='DEFAULT'*,
               partition_label='DEFAULT', *target_project_id='DEFAULT'*,
               running_as_id='DEFAULT')

> Update a container
>> **Parameters**
>>> **id** [integer] The ID for the script.
>>> **name** [string, optional] The name of the container.
>>> **parent_id** [integer, optional] The ID of the parent job that will trigger this script
>>> **user_context** [string, optional] "runner" or "author", who to execute the script as when run as a template.
>>> **params** [list, optional::] A definition of the parameters this script accepts in the arguments field. - name : string
>>>> The variable's name as used within your code.

- **label** [string] The label to present to users when asking them for the value.
- **description** [string] A short sentence or fragment describing this parameter to the end user.
- **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
- **required** [boolean] Whether this param is required.
- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.
- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

arguments [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.

schedule [dict, optional::]
- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

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

required_resources [dict, optional::]
- **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.
- **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.
- **disk_space** [number/float] The amount of disk space, in GB, to allocate for the container. This space will be used to hold the git repo configured for the container and anything your container writes to /tmp

or /data. Fractional values (e.g. 0.25) are supported.

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

repo_ref [string, optional] The tag or branch of the github repo to clone into the container.

remote_host_credential_id [integer, optional] The id of the database credentials to pass into the environment of the container.

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

docker_command [string, optional] The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]. Defaults to the Docker image's ENTRYPOINT/CMD.

docker_image_name [string, optional] The name of the docker image to pull from DockerHub.

docker_image_tag [string, optional] The tag of the docker image to pull from DockerHub.

instance_type [string, optional] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.

cancel_timeout [integer, optional] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

time_zone [string, optional] The time zone of this script.

partition_label [string, optional] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.

target_project_id [integer, optional] Target project to which script outputs will be added.

running_as_id [integer, optional] The ID of the runner of this script.

Returns

civis.response.Response

- **id** [integer] The ID for the script.
- **name** [string] The name of the container.
- **type** [string] The type of the script (e.g Container)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    – **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

  The variable's name as used within your code.

  – **label** [string] The label to present to users when asking them for the value.

  – **description** [string] A short sentence or fragment describing this parameter to the end user.

  – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

  – **required** [boolean] Whether this param is required.

  – **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

  – **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

  – **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template script.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

  – **details** [string] The details link to get more information about the script.

  – **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

  – **scheduled** [boolean] If the item is scheduled.

  – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  – **scheduled_hours** [list] Hours of the day it is scheduled on.

- **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

  - **urls** [list] URLs to receive a POST request at job completion

  - **success_email_subject** [string] Custom subject line for success e-mail.

  - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

  - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

  - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

  - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

  - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

  - **success_on** [boolean] If success email notifications are on.

  - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **required_resources** [dict::]

  - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

  - **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

  - **disk_space** [number/float] The amount of disk space, in GB, to allocate for the container. This space will be used to hold the git repo configured for the container and anything your container writes to /tmp or /data. Fractional values (e.g. 0.25) are supported.

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

- **repo_ref** [string] The tag or branch of the github repo to clone into the container.
- **remote_host_credential_id** [integer] The id of the database credentials to pass into the environment of the container.
- **git_credential_id** [integer] The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.
- **docker_command** [string] The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]. Defaults to the Docker image's ENTRYPOINT/CMD.
- **docker_image_name** [string] The name of the docker image to pull from DockerHub.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.
- **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.
- **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **time_zone** [string] The time zone of this script.
- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **target_project_id** [integer] Target project to which script outputs will be added.
- **running_as_id** [integer] The ID of the runner of this script.

patch_custom(*id*, *, *name='DEFAULT'*, *parent_id='DEFAULT'*, *arguments='DEFAULT'*, *remote_host_id='DEFAULT'*, *credential_id='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*, *time_zone='DEFAULT'*, *target_project_id='DEFAULT'*, *required_resources='DEFAULT'*, *partition_label='DEFAULT'*, *running_as_id='DEFAULT'*)

Update some attributes of this Custom Script

Parameters

id [integer] The ID for the script.

name [string, optional] The name of the script.

parent_id [integer, optional] The ID of the parent job that will trigger this script

arguments [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.

> > **remote_host_id** [integer, optional] The remote host ID that this script will connect to.
> > **credential_id** [integer, optional] The credential that this script will use.
> > **schedule** [dict, optional::]
> >
> > > - **scheduled** [boolean] If the item is scheduled.
> > > - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > > - **scheduled_hours** [list] Hours of the day it is scheduled on.
> > > - **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> > > - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
> > > - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
> >
> > **notifications** [dict, optional::]
> >
> > > - **urls** [list] URLs to receive a POST request at job completion
> > > - **success_email_subject** [string] Custom subject line for success e-mail.
> > > - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.
> > > - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.
> > > - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."
> > > - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.
> > > - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.
> > > - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.
> > > - **success_on** [boolean] If success email notifications are on.
> > > - **failure_on** [boolean] If failure email notifications are on.
> >
> > **time_zone** [string, optional] The time zone of this script.
> > **target_project_id** [integer, optional] Target project to which script outputs will be added.
> > **required_resources** [dict, optional::]
> >
> > > - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares.
> > > - **memory** [integer] The amount of RAM to allocate for the container (in MB).
> > > - **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.
> >
> > **partition_label** [string, optional] The partition label used to run this object. Only applicable for jobs using Docker.Not generally available. Beware this attribute may be removed in the future.
> > **running_as_id** [integer, optional] The ID of the runner of this script.
>
> Returns
>
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID for the script.
> > > - **name** [string] The name of the script.
> > > - **type** [string] The type of the script (e.g Custom)
> > > - **created_at** [string/time] The time this script was created.
> > > - **updated_at** [string/time] The time the script was last updated.
> > > - **author** [dict::]

- **id** [integer] The ID of this user.

- **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** : string
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    - **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template script.
- **ui_report_url** [integer] The url of the custom HTML.
- **ui_report_id** [integer] The id of the report with the custom HTML.
- **ui_report_provide_api_key** [boolean] Whether the ui report requests an API Key from the report viewer.

- **template_script_name** [string] The name of the template script.
- **template_note** [string] The template's note.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **code_preview** [string] The code that this script will run with arguments inserted.
- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

  - **–** id : integer

  - **–** state : string

  - **– created_at** [string/time] The time that the run was queued.

  - **– started_at** [string/time] The time that the run started.

  - **– finished_at** [string/time] The time that the run completed.

  - **– error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **target_project_id** [integer] Target project to which script outputs will be added.
- **last_successful_run** [dict::]

  - **–** id : integer

  - **–** state : string

  - **– created_at** [string/time] The time that the run was queued.

  - **– started_at** [string/time] The time that the run started.

  - **– finished_at** [string/time] The time that the run completed.

  - **– error** [string] The error message for this run, if present.
- **required_resources** [dict::]

  - **– cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares.

  - **– memory** [integer] The amount of RAM to allocate for the container (in MB).

  - **– 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.
- **partition_label** [string] The partition label used to run this object. Only applicable for jobs using Docker.Not generally available. Beware this attribute may be removed in the future.
- **running_as_id** [integer] The ID of the runner of this script.

patch_javascript(*id*, ***, *name='DEFAULT'*, *parent_id='DEFAULT'*, *user_context='DEFAULT'*,
  params='DEFAULT', *arguments='DEFAULT'*, *schedule='DEFAULT'*,
  notifications='DEFAULT', *next_run_at='DEFAULT'*, *time_zone='DEFAULT'*,
  target_project_id='DEFAULT', *source='DEFAULT'*, *remote_host_id='DEFAULT'*,
  credential_id='DEFAULT', *running_as_id='DEFAULT'*)

Update some attributes of this JavaScript Script

Parameters

id [integer] The ID for the script.

name [string, optional] The name of the script.

parent_id [integer, optional] The ID of the parent job that will trigger this script

user_context [string, optional] "runner" or "author", who to execute the script as when
run as a template.

params [list, optional::] A definition of the parameters this script accepts in the argu-
ments field. - name : string

The variable's name as used within your code.

- **label** [string] The label to present to users when asking them for the value.
- **description** [string] A short sentence or fragment describing this param-
eter to the end user.
- **type** [string] The type of parameter. Valid options: string,
multi_line_string, integer, float, bool, file, table, database, cre-
dential_aws, credential_redshift, or credential_custom
- **required** [boolean] Whether this param is required.
- **value** [string] The value you would like to set this param to. Setting this
value makes this parameter a fixed param.
- **default** [string] If an argument for this parameter is not defined, it will
use this default value. Use true, True, t, y, yes, or 1 for true bool's
or false, False, f, n, no, or 0 for false bool's. Cannot be used for
parameters that are required or a credential type.
- **allowed_values** [list] The possible values this parameter can take, effec-
tively making this an enumerable parameter. Allowed values is an
array of hashes of the following format: *{label: 'Import', 'value':
'import'}*

arguments [dict, optional] Parameter-value pairs to use when running this script. Only
settable if this script has defined parameters.

schedule [dict, optional::]
- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting
at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes,
number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on,
mutually exclusive with scheduledDays.

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

next_run_at [string/time, optional] The time of the next scheduled run.

time_zone [string, optional] The time zone of this script.

> **target_project_id** [integer, optional] Target project to which script outputs will be added.
>
> **source** [string, optional] The body/text of the script.
>
> **remote_host_id** [integer, optional] The remote host ID that this script will connect to.
>
> **credential_id** [integer, optional] The credential that this script will use.
>
> **running_as_id** [integer, optional] The ID of the runner of this script.

Returns

> *civis.response.Response*
>
> - **id** [integer] The ID for the script.
> - **name** [string] The name of the script.
> - **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
> - **created_at** [string/time] The time this script was created.
> - **updated_at** [string/time] The time the script was last updated.
> - **author** [dict::]
>
>> - **id** [integer] The ID of this user.
>>
>> - **name** [string] This user's name.
>>
>> - **username** [string] This user's username.
>>
>> - **initials** [string] This user's initials.
>>
>> - **online** [boolean] Whether this user is online.
>
> - **state** [string] The status of the script's last run.
> - **finished_at** [string/time] The time that the script's last run finished.
> - **category** [string] The category of the script.
> - **projects** [list::] A list of projects containing the script. - id : integer
>
>> The ID for the project.
>>
>> - **name** [string] The name of the project.
>
> - **parent_id** [integer] The ID of the parent job that will trigger this script
> - **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
> - **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string
>
>> The variable's name as used within your code.
>>
>> - **label** [string] The label to present to users when asking them for the value.
>>
>> - **description** [string] A short sentence or fragment describing this parameter to the end user.
>>
>> - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
>>
>> - **required** [boolean] Whether this param is required.
>>
>> - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
>>
>> - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t,

> > > y, yes, or 1 for true bool's or false, False, f, n, no, or 0
> > > for false bool's. Cannot be used for parameters that are
> > > required or a credential type.
>
> > – **allowed_values** [list] The possible values this parameter can
> > take, effectively making this an enumerable parameter.
> > Allowed values is an array of hashes of the following
> > format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script.
  Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this
  script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this
  one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

  > – **details** [string] The details link to get more information about
  > the script.
  >
  > – **runs** [string] The runs link to get the run information list for
  > this script.

- **schedule** [dict::]

  > – **scheduled** [boolean] If the item is scheduled.
  >
  > – **scheduled_days** [list] Days of the week, based on numeric
  > value starting at 0 for Sunday. Mutually exclusive with
  > scheduledDaysOfMonth
  >
  > – **scheduled_hours** [list] Hours of the day it is scheduled on.
  >
  > – **scheduled_minutes** [list] Minutes of the day it is scheduled
  > on.
  >
  > – **scheduled_runs_per_hour** [integer] Alternative to sched-
  > uled minutes, number of times to run per hour.
  >
  > – **scheduled_days_of_month** [list] Days of the month it is
  > scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

  > – **urls** [list] URLs to receive a POST request at job completion
  >
  > – **success_email_subject** [string] Custom subject line for suc-
  > cess e-mail.
  >
  > – **success_email_body** [string] Custom body text for success
  > e-mail, written in Markdown.
  >
  > – **success_email_addresses** [list] Addresses to notify by e-mail
  > when the job completes successfully.
  >
  > – **success_email_from_name** [string] Name from which suc-
  > cess emails are sent; defaults to "Civis."
  >
  > – **success_email_reply_to** [string] Address for replies to suc-
  > cess emails; defaults to the author of the job.

- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **source** [string] The body/text of the script.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **running_as_id** [integer] The ID of the runner of this script.

patch_javascript_git(*id*, ***, *git_ref='DEFAULT'*, *git_branch='DEFAULT'*, *git_path='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref_type='DEFAULT'*, *pull_from_git='DEFAULT'*)

Update an attached git file

Parameters

id [integer] The ID of the file.

git_ref [string, optional] A git reference specifying an unambiguous version of the file. Can be a branch name, or the full or shortened SHA of a commit.

git_branch [string, optional] The git branch that the file is on.

git_path [string, optional] The path of the file in the repository.

git_repo_url [string, optional] The URL of the git repository.

git_ref_type [string, optional] Specifies if the file is versioned by branch or tag.

pull_from_git [boolean, optional] Automatically pull latest commit from git. Only works for scripts.

Returns

civis.response.Response

- **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
- **git_branch** [string] The git branch that the file is on.
- **git_path** [string] The path of the file in the repository.
- **git_repo** [dict::]

    – **id** [integer] The ID for this git repository.

    – **repo_url** [string] The URL for this git repository.

    – created_at : string/time

    – updated_at : string/time

- **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
- **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

patch_javascript_runs(*id*, *run_id*, *, *error='DEFAULT'*)

Update the given run

Parameters

id [integer] ID of the Job

run_id [integer] ID of the Run

error [string, optional] The error message to update

Returns

None Response code 204: success

patch_python3(*id*, *, *name='DEFAULT'*, *parent_id='DEFAULT'*, *user_context='DEFAULT'*, *params='DEFAULT'*, *arguments='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*, *next_run_at='DEFAULT'*, *time_zone='DEFAULT'*, *target_project_id='DEFAULT'*, *required_resources='DEFAULT'*, *instance_type='DEFAULT'*, *source='DEFAULT'*, *cancel_timeout='DEFAULT'*, *docker_image_tag='DEFAULT'*, *partition_label='DEFAULT'*, *running_as_id='DEFAULT'*)

Update some attributes of this Python Script

Parameters

id [integer] The ID for the script.

name [string, optional] The name of the script.

parent_id [integer, optional] The ID of the parent job that will trigger this script

user_context [string, optional] "runner" or "author", who to execute the script as when run as a template.

params [list, optional::] A definition of the parameters this script accepts in the arguments field. - name : string

The variable's name as used within your code.

- **label** [string] The label to present to users when asking them for the value.
- **description** [string] A short sentence or fragment describing this parameter to the end user.
- **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
- **required** [boolean] Whether this param is required.
- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's

or false, False, f, n, no, or 0 for false bool's. Cannot be used for
parameters that are required or a credential type.

- **allowed_values** [list] The possible values this parameter can take, effec-
tively making this an enumerable parameter. Allowed values is an
array of hashes of the following format: *{label: 'Import', 'value':
'import'}*

arguments [dict, optional] Parameter-value pairs to use when running this script. Only
settable if this script has defined parameters.

schedule [dict, optional::]

- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting
at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes,
number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on,
mutually exclusive with scheduledDays.

notifications [dict, optional::]

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

next_run_at [string/time, optional] The time of the next scheduled run.

time_zone [string, optional] The time zone of this script.

target_project_id [integer, optional] Target project to which script outputs will be
added.

required_resources [dict, optional::]

- **cpu** [integer] The number of CPU shares to allocate for the container.
Each core has 1000 shares. Must be at least 2 shares.
- **memory** [integer] The amount of RAM to allocate for the container (in
MB). Must be at least 4 MB.
- **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 con-
figured for the container and anything your container writes to /tmp
or /data. Fractional values (e.g. 0.25) are supported.

instance_type [string, optional] The EC2 instance type to deploy to. Only available
for jobs running on kubernetes.

source [string, optional] The body/text of the script.

cancel_timeout [integer, optional] The amount of time (in seconds) to wait before
forcibly terminating the script. When the script is cancelled, it is first sent a
TERM signal. If the script is still running after the timeout, it is sent a KILL

> signal. Defaults to 0.
>
> **docker_image_tag** [string, optional] The tag of the docker image to pull from DockerHub.
>
> **partition_label** [string, optional] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
>
> **running_as_id** [integer, optional] The ID of the runner of this script.

Returns

> *civis.response.Response*
>
> > - **id** [integer] The ID for the script.
> > - **name** [string] The name of the script.
> > - **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
> > - **created_at** [string/time] The time this script was created.
> > - **updated_at** [string/time] The time the script was last updated.
> > - **author** [dict::]
> >
> > > - **id** [integer] The ID of this user.
> > >
> > > - **name** [string] This user's name.
> > >
> > > - **username** [string] This user's username.
> > >
> > > - **initials** [string] This user's initials.
> > >
> > > - **online** [boolean] Whether this user is online.
> >
> > - **state** [string] The status of the script's last run.
> > - **finished_at** [string/time] The time that the script's last run finished.
> > - **category** [string] The category of the script.
> > - **projects** [list::] A list of projects containing the script. - id : integer
> >
> > > The ID for the project.
> >
> > > - **name** [string] The name of the project.
> >
> > - **parent_id** [integer] The ID of the parent job that will trigger this script
> > - **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
> > - **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string
> >
> > > The variable's name as used within your code.
> >
> > > - **label** [string] The label to present to users when asking them for the value.
> > >
> > > - **description** [string] A short sentence or fragment describing this parameter to the end user.
> > >
> > > - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
> > >
> > > - **required** [boolean] Whether this param is required.
> > >
> > > - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
> > >
> > > - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t,

> y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

> – **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

> – **details** [string] The details link to get more information about the script.

> – **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

> – **scheduled** [boolean] If the item is scheduled.

> – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

> – **scheduled_hours** [list] Hours of the day it is scheduled on.

> – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

> – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

> – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

> – **urls** [list] URLs to receive a POST request at job completion

> – **success_email_subject** [string] Custom subject line for success e-mail.

> – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

> – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

> – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

> – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.

- **time_zone** [string] The time zone of this script.

- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

- **hidden** [boolean] The hidden status of the item.

- **target_project_id** [integer] Target project to which script outputs will be added.

- **archived** [string] The archival status of the requested item(s).

- **required_resources** [dict::]

    - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

    - **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

    - **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.

- **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.

- **source** [string] The body/text of the script.

- **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.
- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
- **running_as_id** [integer] The ID of the runner of this script.

patch_python3_git(*id*, ***, *git_ref='DEFAULT'*, *git_branch='DEFAULT'*, *git_path='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref_type='DEFAULT'*, *pull_from_git='DEFAULT'*)

> Update an attached git file
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the file.
> > >
> > > **git_ref** [string, optional] A git reference specifying an unambiguous version of the file. Can be a branch name, or the full or shortened SHA of a commit.
> > >
> > > **git_branch** [string, optional] The git branch that the file is on.
> > >
> > > **git_path** [string, optional] The path of the file in the repository.
> > >
> > > **git_repo_url** [string, optional] The URL of the git repository.
> > >
> > > **git_ref_type** [string, optional] Specifies if the file is versioned by branch or tag.
> > >
> > > **pull_from_git** [boolean, optional] Automatically pull latest commit from git. Only works for scripts.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > - **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
> > > - **git_branch** [string] The git branch that the file is on.
> > > - **git_path** [string] The path of the file in the repository.
> > > - **git_repo** [dict::]
> > >
> > > > - **id** [integer] The ID for this git repository.
> > > >
> > > > - **repo_url** [string] The URL for this git repository.
> > > >
> > > > - created_at : string/time
> > > >
> > > > - updated_at : string/time
> > >
> > > - **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
> > > - **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

patch_python3_runs(*id*, *run_id*, ***, *error='DEFAULT'*)

> Update the given run
>
> > **Parameters**
> >
> > > **id** [integer] ID of the Job
> > >
> > > **run_id** [integer] ID of the Run
> > >
> > > **error** [string, optional] The error message to update
> >
> > **Returns**
> >
> > > **None** Response code 204: success

patch_r(*id*, ***, *name='DEFAULT'*, *parent_id='DEFAULT'*, *user_context='DEFAULT'*, *params='DEFAULT'*, *arguments='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*, *next_run_at='DEFAULT'*, *time_zone='DEFAULT'*, *target_project_id='DEFAULT'*, *required_resources='DEFAULT'*, *instance_type='DEFAULT'*, *source='DEFAULT'*, *cancel_timeout='DEFAULT'*, *docker_image_tag='DEFAULT'*, *partition_label='DEFAULT'*, *running_as_id='DEFAULT'*)

> Update some attributes of this R Script

Parameters

id [integer] The ID for the script.

name [string, optional] The name of the script.

parent_id [integer, optional] The ID of the parent job that will trigger this script

user_context [string, optional] "runner" or "author", who to execute the script as when run as a template.

params [list, optional::] A definition of the parameters this script accepts in the arguments field. - name : string

The variable's name as used within your code.

- **label** [string] The label to present to users when asking them for the value.
- **description** [string] A short sentence or fragment describing this parameter to the end user.
- **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
- **required** [boolean] Whether this param is required.
- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.
- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

arguments [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.

schedule [dict, optional::]

- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

notifications [dict, optional::]

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

- **success_on** [boolean] If success email notifications are on.
- **failure_on** [boolean] If failure email notifications are on.

next_run_at [string/time, optional] The time of the next scheduled run.

time_zone [string, optional] The time zone of this script.

target_project_id [integer, optional] Target project to which script outputs will be added.

required_resources [dict, optional::]

- **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.
- **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.
- **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.

instance_type [string, optional] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.

source [string, optional] The body/text of the script.

cancel_timeout [integer, optional] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

docker_image_tag [string, optional] The tag of the docker image to pull from DockerHub.

partition_label [string, optional] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.

running_as_id [integer, optional] The ID of the runner of this script.

Returns

civis.response.Response

- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    – **name** [string] The name of the project.
- **parent_id** [integer] The ID of the parent job that will trigger this script

- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    - **details** [string] The details link to get more information about the script.

    - **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

- **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

  - **urls** [list] URLs to receive a POST request at job completion

  - **success_email_subject** [string] Custom subject line for success e-mail.

  - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

  - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

  - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

  - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

  - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

  - **success_on** [boolean] If success email notifications are on.

  - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.

- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **required_resources** [dict::]

    – **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

    – **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

    – **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.

- **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.
- **source** [string] The body/text of the script.
- **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.
- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
- **running_as_id** [integer] The ID of the runner of this script.

patch_r_git(*id*, ***, *git_ref='DEFAULT'*, *git_branch='DEFAULT'*, *git_path='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref_type='DEFAULT'*, *pull_from_git='DEFAULT'*)

Update an attached git file

Parameters

id [integer] The ID of the file.

git_ref [string, optional] A git reference specifying an unambiguous version of the file. Can be a branch name, or the full or shortened SHA of a commit.

git_branch [string, optional] The git branch that the file is on.

git_path [string, optional] The path of the file in the repository.

git_repo_url [string, optional] The URL of the git repository.

git_ref_type [string, optional] Specifies if the file is versioned by branch or tag.

pull_from_git [boolean, optional] Automatically pull latest commit from git. Only works for scripts.

Returns

civis.response.Response

- **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
- **git_branch** [string] The git branch that the file is on.
- **git_path** [string] The path of the file in the repository.
- **git_repo** [dict::]

    – **id** [integer] The ID for this git repository.

    – **repo_url** [string] The URL for this git repository.

> > > > – created_at : string/time
> > > >
> > > > – updated_at : string/time
> > >
> > > • **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
> > > • **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

patch_r_runs(*id*, *run_id*, ***, *error='DEFAULT'*)

> Update the given run
>
> > **Parameters**
> >
> > > **id** [integer] ID of the Job
> > > **run_id** [integer] ID of the Run
> > > **error** [string, optional] The error message to update
> >
> > **Returns**
> >
> > > **None** Response code 204: success

patch_sql(*id*, ***, *name='DEFAULT'*, *parent_id='DEFAULT'*, *user_context='DEFAULT'*,
params='DEFAULT', *arguments='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*,
next_run_at='DEFAULT', *time_zone='DEFAULT'*, *target_project_id='DEFAULT'*,
sql='DEFAULT', *remote_host_id='DEFAULT'*, *credential_id='DEFAULT'*,
csv_settings='DEFAULT', *running_as_id='DEFAULT'*)

> Update some attributes of this SQL script
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the script.
> > > **name** [string, optional] The name of the script.
> > > **parent_id** [integer, optional] The ID of the parent job that will trigger this script
> > > **user_context** [string, optional] "runner" or "author", who to execute the script as when run as a template.
> > > **params** [list, optional::] A definition of the parameters this script accepts in the arguments field. - name : string
> > > > The variable's name as used within your code.
> > > >
> > > > • **label** [string] The label to present to users when asking them for the value.
> > > > • **description** [string] A short sentence or fragment describing this parameter to the end user.
> > > > • **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
> > > > • **required** [boolean] Whether this param is required.
> > > > • **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
> > > > • **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.
> > > > • **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*
> > >
> > > **arguments** [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
> > > **schedule** [dict, optional::]
> > > > • **scheduled** [boolean] If the item is scheduled.
> > > > • **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

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

next_run_at [string/time, optional] The time of the next scheduled run.

time_zone [string, optional] The time zone of this script.

target_project_id [integer, optional] Target project to which script outputs will be added.

sql [string, optional] The raw SQL query for the script.

remote_host_id [integer, optional] The remote host ID that this script will connect to.

credential_id [integer, optional] The credential that this script will use.

csv_settings [dict, optional::]
- **include_header** [boolean] Whether or not to include headers in the output data. Default: true
- **compression** [string] The type of compression to use, if any, one of "none", "zip", or "gzip". Default: gzip
- **column_delimiter** [string] Which delimiter to use, one of "comma", "tab", or "pipe". Default: comma
- **unquoted** [boolean] Whether or not to quote fields. Default: false
- **force_multifile** [boolean] Whether or not the csv should be split into multiple files. Default: false
- **filename_prefix** [string] A user specified filename prefix for the output file to have. Default: null
- **max_file_size** [integer] The max file size, in MB, created files will be. Only available when force_multifile is true.

running_as_id [integer, optional] The ID of the runner of this script.

Returns

civis.response.Response
- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

- **id** [integer] The ID of this user.

- **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    - **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.

- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    – **details** [string] The details link to get more information about the script.

    – **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

    – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    – **scheduled_hours** [list] Hours of the day it is scheduled on.

    – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    – **urls** [list] URLs to receive a POST request at job completion

    – **success_email_subject** [string] Custom subject line for success e-mail.

    – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    – **success_on** [boolean] If success email notifications are on.

    – **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    – id : integer

    – state : string

    – **created_at** [string/time] The time that the run was queued.

    – **started_at** [string/time] The time that the run started.

    – **finished_at** [string/time] The time that the run completed.

    – **error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **sql** [string] The raw SQL query for the script.
- **expanded_arguments** [dict] Expanded arguments for use in injecting into different environments.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **code_preview** [string] The code that this script will run with arguments inserted.
- **csv_settings** [dict::]

    – **include_header** [boolean] Whether or not to include headers in the output data. Default: true

    – **compression** [string] The type of compression to use, if any, one of "none", "zip", or "gzip". Default: gzip

    – **column_delimiter** [string] Which delimiter to use, one of "comma", "tab", or "pipe". Default: comma

    – **unquoted** [boolean] Whether or not to quote fields. Default: false

    – **force_multifile** [boolean] Whether or not the csv should be split into multiple files. Default: false

    – **filename_prefix** [string] A user specified filename prefix for the output file to have. Default: null

    – **max_file_size** [integer] The max file size, in MB, created files will be. Only available when force_multifile is true.
- **running_as_id** [integer] The ID of the runner of this script.

patch_sql_git(*id*, *, *git_ref='DEFAULT'*, *git_branch='DEFAULT'*, *git_path='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref_type='DEFAULT'*, *pull_from_git='DEFAULT'*)

Update an attached git file

    Parameters

        id [integer] The ID of the file.

        git_ref [string, optional] A git reference specifying an unambiguous version of the file. Can be a branch name, or the full or shortened SHA of a commit.

        git_branch [string, optional] The git branch that the file is on.

> > > > **git_path** [string, optional] The path of the file in the repository.
> > > > **git_repo_url** [string, optional] The URL of the git repository.
> > > > **git_ref_type** [string, optional] Specifies if the file is versioned by branch or tag.
> > > > **pull_from_git** [boolean, optional] Automatically pull latest commit from git. Only works for scripts.
> > > **Returns**
> > > > *civis.response.Response*
> > > > > • **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
> > > > > • **git_branch** [string] The git branch that the file is on.
> > > > > • **git_path** [string] The path of the file in the repository.
> > > > > • **git_repo** [dict::]
> > > > > > – **id** [integer] The ID for this git repository.
> > > > > > – **repo_url** [string] The URL for this git repository.
> > > > > > – created_at : string/time
> > > > > > – updated_at : string/time
> > > > > • **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
> > > > > • **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

patch_sql_runs(*id*, *run_id*, *, *error='DEFAULT'*)
> > Update the given run
> > > **Parameters**
> > > > **id** [integer] ID of the Job
> > > > **run_id** [integer] ID of the Run
> > > > **error** [string, optional] The error message to update
> > > **Returns**
> > > > **None** Response code 204: success

post(*name*, *remote_host_id*, *credential_id*, *sql*, *, *params='DEFAULT'*, *arguments='DEFAULT'*, *template_script_id='DEFAULT'*, *notifications='DEFAULT'*, *hidden='DEFAULT'*)
> > Create a script
> > > **Parameters**
> > > > **name** [string] The name of the script.
> > > > **remote_host_id** [integer] The database ID.
> > > > **credential_id** [integer] The credential ID.
> > > > **sql** [string] The raw SQL query for the script.
> > > > **params** [list, optional::] A definition of the parameters this script accepts in the arguments field. Cannot be set if this script uses a template script. - name : string The variable's name as used within your code.
> > > > > • **label** [string] The label to present to users when asking them for the value.
> > > > > • **description** [string] A short sentence or fragment describing this parameter to the end user.
> > > > > • **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
> > > > > • **required** [boolean] Whether this param is required.
> > > > > • **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
> > > > > • **default** [string] If an argument for this parameter is not defined, it will

> > > > use this default value. Use true, True, t, y, yes, or 1 for true bool's
> > > > or false, False, f, n, no, or 0 for false bool's. Cannot be used for
> > > > parameters that are required or a credential type.
> > > - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

> > **arguments** [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.

> > **template_script_id** [integer, optional] The ID of the template script, if any. A script cannot both have a template script and be a template for other scripts.

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

> > **hidden** [boolean, optional] The hidden status of the item.

> **Returns**

> *civis.response.Response*
> > > - **id** [integer] The ID for the script.
> > > - **name** [string] The name of the script.
> > > - **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
> > > - **created_at** [string/time] The time this script was created.
> > > - **updated_at** [string/time] The time the script was last updated.
> > > - **author** [dict::]

> > > > - **id** [integer] The ID of this user.

> > > > - **name** [string] This user's name.

> > > > - **username** [string] This user's username.

> > > > - **initials** [string] This user's initials.

> > > > - **online** [boolean] Whether this user is online.

> > > - **state** [string] The status of the script's last run.
> > > - **finished_at** [string/time] The time that the script's last run finished.
> > > - **category** [string] The category of the script.
> > > - **projects** [list::] A list of projects containing the script. - id : integer

> > > > The ID for the project.

> > > > - **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

  The variable's name as used within your code.

  – **label** [string] The label to present to users when asking them for the value.

  – **description** [string] A short sentence or fragment describing this parameter to the end user.

  – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

  – **required** [boolean] Whether this param is required.

  – **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

  – **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

  – **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

  – **details** [string] The details link to get more information about the script.

  – **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

  – **scheduled** [boolean] If the item is scheduled.

  – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  – **scheduled_hours** [list] Hours of the day it is scheduled on.

> > > – **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> >
> > > – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
> >
> > > – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

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

- **running_as** [dict::]

> > > – **id** [integer] The ID of this user.
> >
> > > – **name** [string] This user's name.
> >
> > > – **username** [string] This user's username.
> >
> > > – **initials** [string] This user's initials.
> >
> > > – **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

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

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.

> - **target_project_id** [integer] Target project to which script outputs will be added.
> - **archived** [string] The archival status of the requested item(s).
> - **template_script_id** [integer] The ID of the template script, if any.

post_cancel(*id*)

    Cancel a run

        Parameters

            id [integer] The ID of the job.

        Returns

            civis.response.Response

> - **id** [integer] The ID of the run.
> - **state** [string] The state of the run, one of 'queued', 'running' or 'cancelled'.
> - **is_cancel_requested** [boolean] True if run cancel requested, else false.

post_containers(*required_resources*, *docker_image_name*, ***, *name='DEFAULT'*, *parent_id='DEFAULT'*, *user_context='DEFAULT'*, *params='DEFAULT'*, *arguments='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*, *repo_http_uri='DEFAULT'*, *repo_ref='DEFAULT'*, *remote_host_credential_id='DEFAULT'*, *git_credential_id='DEFAULT'*, *docker_command='DEFAULT'*, *docker_image_tag='DEFAULT'*, *instance_type='DEFAULT'*, *cancel_timeout='DEFAULT'*, *time_zone='DEFAULT'*, *partition_label='DEFAULT'*, *hidden='DEFAULT'*, *target_project_id='DEFAULT'*, *running_as_id='DEFAULT'*)

    Create a container

        Parameters

            required_resources [dict::]

> - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.
> - **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.
> - **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_image_name [string] The name of the docker image to pull from DockerHub.

            name [string, optional] The name of the container.

            parent_id [integer, optional] The ID of the parent job that will trigger this script

            user_context [string, optional] "runner" or "author", who to execute the script as when run as a template.

            params [list, optional::] A definition of the parameters this script accepts in the arguments field. - name : string

                The variable's name as used within your code.

> - **label** [string] The label to present to users when asking them for the value.
> - **description** [string] A short sentence or fragment describing this parameter to the end user.
> - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
> - **required** [boolean] Whether this param is required.
> - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
> - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's

---

> > > or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.
> > > - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

> **arguments** [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.

> **schedule** [dict, optional::]
> > - **scheduled** [boolean] If the item is scheduled.
> > - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > - **scheduled_hours** [list] Hours of the day it is scheduled on.
> > - **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> > - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
> > - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

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

> **repo_http_uri** [string, optional] The location of a github repo to clone into the container, e.g. github.com/my-user/my-repo.git.

> **repo_ref** [string, optional] The tag or branch of the github repo to clone into the container.

> **remote_host_credential_id** [integer, optional] The id of the database credentials to pass into the environment of the container.

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

> **docker_command** [string, optional] The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]. Defaults to the Docker image's ENTRYPOINT/CMD.

> **docker_image_tag** [string, optional] The tag of the docker image to pull from DockerHub.

> **instance_type** [string, optional] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.

> **cancel_timeout** [integer, optional] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a

> > TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.
>
> **time_zone** [string, optional] The time zone of this script.
>
> **partition_label** [string, optional] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
>
> **hidden** [boolean, optional] The hidden status of the item.
>
> **target_project_id** [integer, optional] Target project to which script outputs will be added.
>
> **running_as_id** [integer, optional] The ID of the runner of this script.
>
> **Returns**
>
> > [*civis.response.Response*](#)
> >
> > > - **id** [integer] The ID for the script.
> > > - **name** [string] The name of the container.
> > > - **type** [string] The type of the script (e.g Container)
> > > - **created_at** [string/time] The time this script was created.
> > > - **updated_at** [string/time] The time the script was last updated.
> > > - **author** [dict::]
> > >
> > > > - **id** [integer] The ID of this user.
> > > >
> > > > - **name** [string] This user's name.
> > > >
> > > > - **username** [string] This user's username.
> > > >
> > > > - **initials** [string] This user's initials.
> > > >
> > > > - **online** [boolean] Whether this user is online.
> > >
> > > - **state** [string] The status of the script's last run.
> > > - **finished_at** [string/time] The time that the script's last run finished.
> > > - **category** [string] The category of the script.
> > > - **projects** [list::] A list of projects containing the script. - id : integer
> > >
> > > > The ID for the project.
> > > >
> > > > - **name** [string] The name of the project.
> > >
> > > - **parent_id** [integer] The ID of the parent job that will trigger this script
> > > - **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
> > > - **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string
> > >
> > > > The variable's name as used within your code.
> > > >
> > > > - **label** [string] The label to present to users when asking them for the value.
> > > >
> > > > - **description** [string] A short sentence or fragment describing this parameter to the end user.
> > > >
> > > > - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
> > > >
> > > > - **required** [boolean] Whether this param is required.
> > > >
> > > > - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

> - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.
>
> - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template script.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

  > - **details** [string] The details link to get more information about the script.
  >
  > - **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

  > - **scheduled** [boolean] If the item is scheduled.
  >
  > - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
  >
  > - **scheduled_hours** [list] Hours of the day it is scheduled on.
  >
  > - **scheduled_minutes** [list] Minutes of the day it is scheduled on.
  >
  > - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
  >
  > - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

  > - **urls** [list] URLs to receive a POST request at job completion
  >
  > - **success_email_subject** [string] Custom subject line for success e-mail.
  >
  > - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.
  >
  > - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.
  >
  > - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

* – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

* – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

* – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

* – **success_on** [boolean] If success email notifications are on.

* – **failure_on** [boolean] If failure email notifications are on.

* **running_as** [dict::]

    * – **id** [integer] The ID of this user.

    * – **name** [string] This user's name.

    * – **username** [string] This user's username.

    * – **initials** [string] This user's initials.

    * – **online** [boolean] Whether this user is online.

* **required_resources** [dict::]

    * – **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

    * – **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

    * – **disk_space** [number/float] The amount of disk space, in GB, to allocate for the container. This space will be used to hold the git repo configured for the container and anything your container writes to /tmp or /data. Fractional values (e.g. 0.25) are supported.

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

* **repo_ref** [string] The tag or branch of the github repo to clone into the container.

* **remote_host_credential_id** [integer] The id of the database credentials to pass into the environment of the container.

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

* **docker_command** [string] The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]. Defaults to the Docker image's ENTRYPOINT/CMD.

* **docker_image_name** [string] The name of the docker image to pull from DockerHub.

* **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.

* **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.

* **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **time_zone** [string] The time zone of this script.
- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **target_project_id** [integer] Target project to which script outputs will be added.
- **running_as_id** [integer] The ID of the runner of this script.

post_containers_clone(*id*, ***, *clone_schedule='DEFAULT'*, *clone_triggers='DEFAULT'*, *clone_notifications='DEFAULT'*)

> Clone this Container Script
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the script.
> > >
> > > **clone_schedule** [boolean, optional] If true, also copy the schedule to the new script.
> > >
> > > **clone_triggers** [boolean, optional] If true, also copy the triggers to the new script.
> > >
> > > **clone_notifications** [boolean, optional] If true, also copy the notifications to the new script.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > - **id** [integer] The ID for the script.
> > > - **name** [string] The name of the container.
> > > - **type** [string] The type of the script (e.g Container)
> > > - **created_at** [string/time] The time this script was created.
> > > - **updated_at** [string/time] The time the script was last updated.
> > > - **author** [dict::]
> > >
> > >   - **id** [integer] The ID of this user.
> > >
> > >   - **name** [string] This user's name.
> > >
> > >   - **username** [string] This user's username.
> > >
> > >   - **initials** [string] This user's initials.
> > >
> > >   - **online** [boolean] Whether this user is online.
> > > - **state** [string] The status of the script's last run.
> > > - **finished_at** [string/time] The time that the script's last run finished.
> > > - **category** [string] The category of the script.
> > > - **projects** [list::] A list of projects containing the script. - id : integer
> > >
> > >   The ID for the project.
> > >
> > >   - **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template script.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    - **details** [string] The details link to get more information about the script.

    - **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

- **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **required_resources** [dict::]

    - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

    - **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

    - **disk_space** [number/float] The amount of disk space, in GB, to allocate for the container. This space will be used to hold the git repo configured for the container and anything your container writes to /tmp or /data. Fractional values (e.g. 0.25) are supported.

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

- **repo_ref** [string] The tag or branch of the github repo to clone into the container.
- **remote_host_credential_id** [integer] The id of the database credentials to pass into the environment of the container.
- **git_credential_id** [integer] The id of the git credential to be used when checking out the specified git repo. If not supplied, the first git credential you've submitted will be used. Unnecessary if no git repo is specified or the git repo is public.
- **docker_command** [string] The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]. Defaults to the Docker image's ENTRYPOINT/CMD.
- **docker_image_name** [string] The name of the docker image to pull from DockerHub.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.
- **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.
- **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **time_zone** [string] The time zone of this script.
- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **target_project_id** [integer] Target project to which script outputs will be added.
- **running_as_id** [integer] The ID of the runner of this script.

post_containers_runs(*id*)

> Start a run
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the container.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > - **id** [integer] The ID of the run.
> > > - **container_id** [integer] The ID of the container.
> > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.

- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.
- **error** [string] The error, if any, returned by the run.
- **max_memory_usage** [number/float] If the run has finished, the maximum amount of memory used during the run, in MB.
- **max_cpu_usage** [number/float] If the run has finished, the maximum amount of cpu used during the run, in millicores.

post_containers_runs_logs(*id*, *run_id*, *, *message='DEFAULT'*, *level='DEFAULT'*, *messages='DEFAULT'*, *child_job_id='DEFAULT'*)

    Add log messages

        Parameters

            id [integer] The ID of the script.

            run_id [integer] The ID of the script run.

            message [string, optional] The log message to store.

            level [string, optional] The log level of this message [default: info]

            messages [list, optional::] If specified, a batch of logs to store. If createdAt timestamps for the logs are supplied, the ordering of this list is not preserved, and the timestamps are used to sort the logs.If createdAt timestamps are not supplied, the ordering of this list is preserved and the logs are given the timestamp of when they were received. - message : string

                The log message to store.

- **level** [string] The log level of this message [default: info]
- **created_at** [string/date-time] The timestamp of this message in ISO 8601 format. This is what logs are ordered by, so it is recommended to use timestamps with nanosecond precision. If absent, defaults to the time that the log was received by the API.

            child_job_id [integer, optional] The ID of the child job the message came from.

        Returns

            None Response code 204: success

post_containers_runs_outputs(*id*, *run_id*, *object_type*, *object_id*)

    Add an output for a run

        Parameters

            id [integer] The ID of the container script.

            run_id [integer] The ID of the run.

            object_type [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue

            object_id [integer] The ID of the output.

         Returns

            civis.response.Response

- **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
- **object_id** [integer] The ID of the output.
- **name** [string] The name of the output.
- **link** [string] The hypermedia link to the output.
- value : string

post_custom(*from_template_id*, *, *name='DEFAULT'*, *parent_id='DEFAULT'*, *arguments='DEFAULT'*, *remote_host_id='DEFAULT'*, *credential_id='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*, *time_zone='DEFAULT'*, *hidden='DEFAULT'*, *target_project_id='DEFAULT'*, *required_resources='DEFAULT'*, *partition_label='DEFAULT'*, *running_as_id='DEFAULT'*)

    Create a Custom Script

        Parameters

> **from_template_id** [integer] The ID of the template script.
>
> **name** [string, optional] The name of the script.
>
> **parent_id** [integer, optional] The ID of the parent job that will trigger this script
>
> **arguments** [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
>
> **remote_host_id** [integer, optional] The remote host ID that this script will connect to.
>
> **credential_id** [integer, optional] The credential that this script will use.
>
> **schedule** [dict, optional::]
>
> > - **scheduled** [boolean] If the item is scheduled.
> > - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > - **scheduled_hours** [list] Hours of the day it is scheduled on.
> > - **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> > - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
> > - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
>
> **notifications** [dict, optional::]
>
> > - **urls** [list] URLs to receive a POST request at job completion
> > - **success_email_subject** [string] Custom subject line for success e-mail.
> > - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.
> > - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.
> > - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."
> > - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.
> > - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.
> > - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.
> > - **success_on** [boolean] If success email notifications are on.
> > - **failure_on** [boolean] If failure email notifications are on.
>
> **time_zone** [string, optional] The time zone of this script.
>
> **hidden** [boolean, optional] The hidden status of the item.
>
> **target_project_id** [integer, optional] Target project to which script outputs will be added.
>
> **required_resources** [dict, optional::]
>
> > - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares.
> > - **memory** [integer] The amount of RAM to allocate for the container (in MB).
> > - **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.
>
> **partition_label** [string, optional] The partition label used to run this object. Only applicable for jobs using Docker.Not generally available. Beware this attribute may be removed in the future.
>
> **running_as_id** [integer, optional] The ID of the runner of this script.
>
> Returns
>
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID for the script.

- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g Custom)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- category : string
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    – **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    – **label** [string] The label to present to users when asking them for the value.

    – **description** [string] A short sentence or fragment describing this parameter to the end user.

    – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    – **required** [boolean] Whether this param is required.

    – **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    – **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    – **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.

- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template script.
- **ui_report_url** [integer] The url of the custom HTML.
- **ui_report_id** [integer] The id of the report with the custom HTML.
- **ui_report_provide_api_key** [boolean] Whether the ui report requests an API Key from the report viewer.
- **template_script_name** [string] The name of the template script.
- **template_note** [string] The template's note.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **code_preview** [string] The code that this script will run with arguments inserted.
- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

    – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    – **scheduled_hours** [list] Hours of the day it is scheduled on.

    – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    – **urls** [list] URLs to receive a POST request at job completion

    – **success_email_subject** [string] Custom subject line for success e-mail.

    – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    – **success_on** [boolean] If success email notifications are on.

    – **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

- **id** [integer] The ID of this user.

- **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **target_project_id** [integer] Target project to which script outputs will be added.
- **last_successful_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.
- **required_resources** [dict::]

  - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares.

  - **memory** [integer] The amount of RAM to allocate for the container (in MB).

  - **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.
- **partition_label** [string] The partition label used to run this object. Only applicable for jobs using Docker.Not generally available. Beware this attribute may be removed in the future.
- **running_as_id** [integer] The ID of the runner of this script.

post_custom_clone(*id*, *, *clone_schedule='DEFAULT'*, *clone_triggers='DEFAULT'*,
    clone_notifications='DEFAULT')

Clone this Custom Script
    Parameters

id [integer] The ID for the script.

clone_schedule [boolean, optional] If true, also copy the schedule to the new script.

clone_triggers [boolean, optional] If true, also copy the triggers to the new script.

clone_notifications [boolean, optional] If true, also copy the notifications to the new script.

Returns

civis.response.Response

- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g Custom)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- category : string
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    - **name** [string] The name of the project.
- **parent_id** [integer] The ID of the parent job that will trigger this script
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template script.
- **ui_report_url** [integer] The url of the custom HTML.
- **ui_report_id** [integer] The id of the report with the custom HTML.
- **ui_report_provide_api_key** [boolean] Whether the ui report requests an API Key from the report viewer.
- **template_script_name** [string] The name of the template script.
- **template_note** [string] The template's note.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **code_preview** [string] The code that this script will run with arguments inserted.
- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **target_project_id** [integer] Target project to which script outputs will be added.
- **last_successful_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **required_resources** [dict::]

    - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares.

    - **memory** [integer] The amount of RAM to allocate for the container (in MB).

    - **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.

- **partition_label** [string] The partition label used to run this object. Only applicable for jobs using Docker.Not generally available. Beware this attribute may be removed in the future.
- **running_as_id** [integer] The ID of the runner of this script.

post_custom_runs(*id*)

Start a run

Parameters

id [integer] The ID of the custom.

Returns

civis.response.Response

- **id** [integer] The ID of the run.
- **custom_id** [integer] The ID of the custom.
- **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **is_cancel_requested** [boolean] True if run cancel requested, else false.
- **started_at** [string/time] The time the last run started at.
- **finished_at** [string/time] The time the last run completed.
- **error** [string] The error, if any, returned by the run.
- **max_memory_usage** [number/float] If the run has finished, the maximum amount of memory used during the run, in MB. Only available if the backing script is a Python, R, or container script.
- **max_cpu_usage** [number/float] If the run has finished, the maximum amount of cpu used during the run, in millicores. Only available if the backing script is a Python, R, or container script.

post_custom_runs_outputs(*id*, *run_id*, *object_type*, *object_id*)

Add an output for a run

Parameters

id [integer] The ID of the custom script.

run_id [integer] The ID of the run.

object_type [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue

object_id [integer] The ID of the output.

Returns

civis.response.Response

- **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
- **object_id** [integer] The ID of the output.
- **name** [string] The name of the output.
- **link** [string] The hypermedia link to the output.
- value : string

post_javascript(*name*, *source*, *remote_host_id*, *credential_id*, ***, *parent_id='DEFAULT'*, *user_context='DEFAULT'*, *params='DEFAULT'*, *arguments='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*, *next_run_at='DEFAULT'*, *time_zone='DEFAULT'*, *hidden='DEFAULT'*, *target_project_id='DEFAULT'*, *running_as_id='DEFAULT'*)

Create a JavaScript Script

Parameters

name [string] The name of the script.

source [string] The body/text of the script.

remote_host_id [integer] The remote host ID that this script will connect to.

credential_id [integer] The credential that this script will use.

parent_id [integer, optional] The ID of the parent job that will trigger this script

user_context [string, optional] "runner" or "author", who to execute the script as when run as a template.

params [list, optional::] A definition of the parameters this script accepts in the arguments field. - name : string

> The variable's name as used within your code.

- **label** [string] The label to present to users when asking them for the value.
- **description** [string] A short sentence or fragment describing this parameter to the end user.
- **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
- **required** [boolean] Whether this param is required.
- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.
- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

arguments [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.

schedule [dict, optional::]
- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

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

next_run_at [string/time, optional] The time of the next scheduled run.

time_zone [string, optional] The time zone of this script.

> > **hidden** [boolean, optional] The hidden status of the item.
> >
> > **target_project_id** [integer, optional] Target project to which script outputs will be added.
> >
> > **running_as_id** [integer, optional] The ID of the runner of this script.
>
> **Returns**
>
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID for the script.
> > > - **name** [string] The name of the script.
> > > - **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
> > > - **created_at** [string/time] The time this script was created.
> > > - **updated_at** [string/time] The time the script was last updated.
> > > - **author** [dict::]
> > >
> > > > - **id** [integer] The ID of this user.
> > > >
> > > > - **name** [string] This user's name.
> > > >
> > > > - **username** [string] This user's username.
> > > >
> > > > - **initials** [string] This user's initials.
> > > >
> > > > - **online** [boolean] Whether this user is online.
> > >
> > > - **state** [string] The status of the script's last run.
> > > - **finished_at** [string/time] The time that the script's last run finished.
> > > - **category** [string] The category of the script.
> > > - **projects** [list::] A list of projects containing the script. - id : integer
> > >
> > > > The ID for the project.
> > > >
> > > > - **name** [string] The name of the project.
> > >
> > > - **parent_id** [integer] The ID of the parent job that will trigger this script
> > > - **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
> > > - **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string
> > >
> > > > The variable's name as used within your code.
> > > >
> > > > - **label** [string] The label to present to users when asking them for the value.
> > > >
> > > > - **description** [string] A short sentence or fragment describing this parameter to the end user.
> > > >
> > > > - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
> > > >
> > > > - **required** [boolean] Whether this param is required.
> > > >
> > > > - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
> > > >
> > > > - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0

> > for false bool's. Cannot be used for parameters that are required or a credential type.

> > – **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

> > – **details** [string] The details link to get more information about the script.

> > – **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

> > – **scheduled** [boolean] If the item is scheduled.

> > – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

> > – **scheduled_hours** [list] Hours of the day it is scheduled on.

> > – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

> > – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

> > – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

> > – **urls** [list] URLs to receive a POST request at job completion

> > – **success_email_subject** [string] Custom subject line for success e-mail.

> > – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

> > – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

> > – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

> > – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

- – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- – **success_on** [boolean] If success email notifications are on.

- – **failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]

    - – **id** [integer] The ID of this user.

    - – **name** [string] This user's name.

    - – **username** [string] This user's username.

    - – **initials** [string] This user's initials.

    - – **online** [boolean] Whether this user is online.
- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - – id : integer

    - – state : string

    - – **created_at** [string/time] The time that the run was queued.

    - – **started_at** [string/time] The time that the run started.

    - – **finished_at** [string/time] The time that the run completed.

    - – **error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **source** [string] The body/text of the script.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **running_as_id** [integer] The ID of the runner of this script.

post_javascript_clone(*id*, *, *clone_schedule='DEFAULT'*, *clone_triggers='DEFAULT'*, *clone_notifications='DEFAULT'*)

> Clone this JavaScript Script
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the script.
> > > **clone_schedule** [boolean, optional] If true, also copy the schedule to the new script.
> > > **clone_triggers** [boolean, optional] If true, also copy the triggers to the new script.
> > > **clone_notifications** [boolean, optional] If true, also copy the notifications to the new script.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > - **id** [integer] The ID for the script.
> > - **name** [string] The name of the script.

- **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    - **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.

- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    – **details** [string] The details link to get more information about the script.

    – **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

    – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    – **scheduled_hours** [list] Hours of the day it is scheduled on.

    – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    – **urls** [list] URLs to receive a POST request at job completion

    – **success_email_subject** [string] Custom subject line for success e-mail.

    – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    – **success_on** [boolean] If success email notifications are on.

    – **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    – **id** [integer] The ID of this user.

> > > > – **name** [string] This user's name.
> > > >
> > > > – **username** [string] This user's username.
> > > >
> > > > – **initials** [string] This user's initials.
> > > >
> > > > – **online** [boolean] Whether this user is online.
> > >
> > > • **next_run_at** [string/time] The time of the next scheduled run.
> > > • **time_zone** [string] The time zone of this script.
> > > • **last_run** [dict::]
> > >
> > > > – id : integer
> > > >
> > > > – state : string
> > > >
> > > > – **created_at** [string/time] The time that the run was queued.
> > > >
> > > > – **started_at** [string/time] The time that the run started.
> > > >
> > > > – **finished_at** [string/time] The time that the run completed.
> > > >
> > > > – **error** [string] The error message for this run, if present.
> > >
> > > • **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
> > > • **hidden** [boolean] The hidden status of the item.
> > > • **target_project_id** [integer] Target project to which script outputs will be added.
> > > • **archived** [string] The archival status of the requested item(s).
> > > • **source** [string] The body/text of the script.
> > > • **remote_host_id** [integer] The remote host ID that this script will connect to.
> > > • **credential_id** [integer] The credential that this script will use.
> > > • **running_as_id** [integer] The ID of the runner of this script.

post_javascript_git_checkout(*id*)

> Checkout content that the existing git_ref points to and save to the object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the file.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **content** [string] The file's contents.
> > > > • **type** [string] The file's type.
> > > > • **size** [integer] The file's size.
> > > > • **file_hash** [string] The SHA of the file.

post_javascript_git_checkout_latest(*id*)

> Checkout latest commit on the current branch of a script or workflow
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the file.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **content** [string] The file's contents.
> > > > • **type** [string] The file's type.
> > > > • **size** [integer] The file's size.
> > > > • **file_hash** [string] The SHA of the file.

post_javascript_git_commits(*id*, *content*, *message*, *file_hash*)

> Commit and push a new version of the file
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the file.

> > **content** [string] The contents to commit to the file.
> > **message** [string] A commit message describing the changes being made.
> > **file_hash** [string] The full SHA of the file being replaced.
> > **Returns**
> > > *civis.response.Response*
> > > > - **content** [string] The file's contents.
> > > > - **type** [string] The file's type.
> > > > - **size** [integer] The file's size.
> > > > - **file_hash** [string] The SHA of the file.

post_javascript_runs(*id*)

> Start a run
> > **Parameters**
> > > **id** [integer] The ID of the javascript.
> > **Returns**
> > > *civis.response.Response*
> > > > - **id** [integer] The ID of the run.
> > > > - **javascript_id** [integer] The ID of the javascript.
> > > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > > - **started_at** [string/time] The time the last run started at.
> > > > - **finished_at** [string/time] The time the last run completed.
> > > > - **error** [string] The error, if any, returned by the run.

post_javascript_runs_outputs(*id*, *run_id*, *object_type*, *object_id*)

> Add an output for a run
> > **Parameters**
> > > **id** [integer] The ID of the javascript script.
> > > **run_id** [integer] The ID of the run.
> > > **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
> > > **object_id** [integer] The ID of the output.
> > **Returns**
> > > *civis.response.Response*
> > > > - **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
> > > > - **object_id** [integer] The ID of the output.
> > > > - **name** [string] The name of the output.
> > > > - **link** [string] The hypermedia link to the output.
> > > > - value : string

post_python3(*name*, *source*, *, *parent_id='DEFAULT'*, *user_context='DEFAULT'*, *params='DEFAULT'*, *arguments='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*, *next_run_at='DEFAULT'*, *time_zone='DEFAULT'*, *hidden='DEFAULT'*, *target_project_id='DEFAULT'*, *required_resources='DEFAULT'*, *instance_type='DEFAULT'*, *cancel_timeout='DEFAULT'*, *docker_image_tag='DEFAULT'*, *partition_label='DEFAULT'*, *running_as_id='DEFAULT'*)

> Create a Python Script
> > **Parameters**
> > > **name** [string] The name of the script.
> > > **source** [string] The body/text of the script.
> > > **parent_id** [integer, optional] The ID of the parent job that will trigger this script
> > > **user_context** [string, optional] "runner" or "author", who to execute the script as when run as a template.

params [list, optional::] A definition of the parameters this script accepts in the arguments field. - name : string

> The variable's name as used within your code.

> - **label** [string] The label to present to users when asking them for the value.
> - **description** [string] A short sentence or fragment describing this parameter to the end user.
> - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
> - **required** [boolean] Whether this param is required.
> - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
> - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.
> - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

arguments [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.

schedule [dict, optional::]
> - **scheduled** [boolean] If the item is scheduled.
> - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> - **scheduled_hours** [list] Hours of the day it is scheduled on.
> - **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
> - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

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

next_run_at [string/time, optional] The time of the next scheduled run.

time_zone [string, optional] The time zone of this script.

hidden [boolean, optional] The hidden status of the item.

target_project_id [integer, optional] Target project to which script outputs will be

---

added.

required_resources [dict, optional::]

- **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.
- **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.
- **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.

instance_type [string, optional] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.

cancel_timeout [integer, optional] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

docker_image_tag [string, optional] The tag of the docker image to pull from DockerHub.

partition_label [string, optional] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.

running_as_id [integer, optional] The ID of the runner of this script.

Returns

civis.response.Response

- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

  – **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

> > > – **label** [string] The label to present to users when asking them
> > > for the value.

> > > – **description** [string] A short sentence or fragment describing
> > > this parameter to the end user.

> > > – **type** [string] The type of parameter. Valid options:
> > > string, multi_line_string, integer, float, bool, file, table,
> > > database, credential_aws, credential_redshift, or creden-
> > > tial_custom

> > > – **required** [boolean] Whether this param is required.

> > > – **value** [string] The value you would like to set this param to.
> > > Setting this value makes this parameter a fixed param.

> > > – **default** [string] If an argument for this parameter is not de-
> > > fined, 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.

> > > – **allowed_values** [list] The possible values this parameter can
> > > take, effectively making this an enumerable parameter.
> > > Allowed values is an array of hashes of the following
> > > format: *{label: 'Import', 'value': 'import'}*

> - **arguments** [dict] Parameter-value pairs to use when running this script.
>   Only settable if this script has defined parameters.
> - **is_template** [boolean] Whether others scripts use this one as a template.
> - **published_as_template_id** [integer] The ID of the template that this
>   script is backing.
> - **from_template_id** [integer] The ID of the template this script uses, if any.
> - **template_dependents_count** [integer] How many other scripts use this
>   one as a template.
> - **template_script_name** [string] The name of the template script.
> - **links** [dict::]

> > > – **details** [string] The details link to get more information about
> > > the script.

> > > – **runs** [string] The runs link to get the run information list for
> > > this script.

> - **schedule** [dict::]

> > > – **scheduled** [boolean] If the item is scheduled.

> > > – **scheduled_days** [list] Days of the week, based on numeric
> > > value starting at 0 for Sunday. Mutually exclusive with
> > > scheduledDaysOfMonth

> > > – **scheduled_hours** [list] Hours of the day it is scheduled on.

> > > – **scheduled_minutes** [list] Minutes of the day it is scheduled
> > > on.

> > > – **scheduled_runs_per_hour** [integer] Alternative to sched-
> > > uled minutes, number of times to run per hour.

> > > – **scheduled_days_of_month** [list] Days of the month it is
> > > scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **required_resources** [dict::]

    - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

> > > > – **memory** [integer] The amount of RAM to allocate for the
> > > > container (in MB). Must be at least 4 MB.
> > > >
> > > > – **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 any-
> > > > thing your container writes to /tmp or /data. Fractional
> > > > values (e.g. 0.25) are supported.

> > > - **instance_type** [string] The EC2 instance type to deploy to. Only available
> > > for jobs running on kubernetes.
> > > - **source** [string] The body/text of the script.
> > > - **cancel_timeout** [integer] The amount of time (in seconds) to wait before
> > > forcibly terminating the script. When the script is cancelled, it is first
> > > sent a TERM signal. If the script is still running after the timeout, it
> > > is sent a KILL signal. Defaults to 0.
> > > - **docker_image_tag** [string] The tag of the docker image to pull from
> > > DockerHub.
> > > - **partition_label** [string] The partition label used to run this object. Not
> > > generally available. Beware this attribute may be removed in the
> > > future.
> > > - **running_as_id** [integer] The ID of the runner of this script.

post_python3_clone(*id*, ***, *clone_schedule='DEFAULT'*, *clone_triggers='DEFAULT'*,
> > *clone_notifications='DEFAULT'*)

> Clone this Python Script

> > **Parameters**

> > > **id** [integer] The ID for the script.
> > > **clone_schedule** [boolean, optional] If true, also copy the schedule to the new script.
> > > **clone_triggers** [boolean, optional] If true, also copy the triggers to the new script.
> > > **clone_notifications** [boolean, optional] If true, also copy the notifications to the new
> > > script.

> > **Returns**

> > *civis.response.Response*

> > > - **id** [integer] The ID for the script.
> > > - **name** [string] The name of the script.
> > > - **type** [string] The type of the script (e.g SQL, Container, Python, R,
> > > JavaScript)
> > > - **created_at** [string/time] The time this script was created.
> > > - **updated_at** [string/time] The time the script was last updated.
> > > - **author** [dict::]
> > >
> > > > – **id** [integer] The ID of this user.
> > > >
> > > > – **name** [string] This user's name.
> > > >
> > > > – **username** [string] This user's username.
> > > >
> > > > – **initials** [string] This user's initials.
> > > >
> > > > – **online** [boolean] Whether this user is online.
> > >
> > > - **state** [string] The status of the script's last run.
> > > - **finished_at** [string/time] The time that the script's last run finished.
> > > - **category** [string] The category of the script.
> > > - **projects** [list::] A list of projects containing the script. - id : integer
> > >
> > > > The ID for the project.
> > >
> > > > – **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    - **details** [string] The details link to get more information about the script.

    - **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

- **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.

- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **required_resources** [dict::]

    - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

    - **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

    - **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.

- **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.
- **source** [string] The body/text of the script.
- **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.
- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
- **running_as_id** [integer] The ID of the runner of this script.

post_python3_git_checkout(*id*)

Checkout content that the existing git_ref points to and save to the object

Parameters

id [integer] The ID of the file.

Returns

civis.response.Response

- **content** [string] The file's contents.
- **type** [string] The file's type.
- **size** [integer] The file's size.
- **file_hash** [string] The SHA of the file.

post_python3_git_checkout_latest(*id*)

Checkout latest commit on the current branch of a script or workflow

Parameters

id [integer] The ID of the file.

Returns

civis.response.Response

- **content** [string] The file's contents.
- **type** [string] The file's type.
- **size** [integer] The file's size.
- **file_hash** [string] The SHA of the file.

post_python3_git_commits(*id*, *content*, *message*, *file_hash*)

Commit and push a new version of the file

Parameters

id [integer] The ID of the file.

> > > **content** [string] The contents to commit to the file.
> > > **message** [string] A commit message describing the changes being made.
> > > **file_hash** [string] The full SHA of the file being replaced.
> >
> > **Returns**
> > > *civis.response.Response*
> > > > - **content** [string] The file's contents.
> > > > - **type** [string] The file's type.
> > > > - **size** [integer] The file's size.
> > > > - **file_hash** [string] The SHA of the file.

post_python3_runs(*id*)

> Start a run
> > **Parameters**
> > > **id** [integer] The ID of the python.
> >
> > **Returns**
> > > *civis.response.Response*
> > > > - **id** [integer] The ID of the run.
> > > > - **python_id** [integer] The ID of the python.
> > > > - **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> > > > - **is_cancel_requested** [boolean] True if run cancel requested, else false.
> > > > - **started_at** [string/time] The time the last run started at.
> > > > - **finished_at** [string/time] The time the last run completed.
> > > > - **error** [string] The error, if any, returned by the run.
> > > > - **max_memory_usage** [number/float] If the run has finished, the maximum amount of memory used during the run, in MB.
> > > > - **max_cpu_usage** [number/float] If the run has finished, the maximum amount of cpu used during the run, in millicores.

post_python3_runs_outputs(*id*, *run_id*, *object_type*, *object_id*)

> Add an output for a run
> > **Parameters**
> > > **id** [integer] The ID of the python script.
> > > **run_id** [integer] The ID of the run.
> > > **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
> > > **object_id** [integer] The ID of the output.
> >
> > **Returns**
> > > *civis.response.Response*
> > > > - **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue
> > > > - **object_id** [integer] The ID of the output.
> > > > - **name** [string] The name of the output.
> > > > - **link** [string] The hypermedia link to the output.
> > > > - value : string

post_r(*name*, *source*, ***, *parent_id='DEFAULT'*, *user_context='DEFAULT'*, *params='DEFAULT'*, *arguments='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*, *next_run_at='DEFAULT'*, *time_zone='DEFAULT'*, *hidden='DEFAULT'*, *target_project_id='DEFAULT'*, *required_resources='DEFAULT'*, *instance_type='DEFAULT'*, *cancel_timeout='DEFAULT'*, *docker_image_tag='DEFAULT'*, *partition_label='DEFAULT'*, *running_as_id='DEFAULT'*)

> Create an R Script
> > **Parameters**
> > > **name** [string] The name of the script.
> > > **source** [string] The body/text of the script.

parent_id [integer, optional] The ID of the parent job that will trigger this script

user_context [string, optional] "runner" or "author", who to execute the script as when run as a template.

params [list, optional::] A definition of the parameters this script accepts in the arguments field. - name : string

> The variable's name as used within your code.

- **label** [string] The label to present to users when asking them for the value.
- **description** [string] A short sentence or fragment describing this parameter to the end user.
- **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
- **required** [boolean] Whether this param is required.
- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.
- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

arguments [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.

schedule [dict, optional::]

- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

notifications [dict, optional::]

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

next_run_at [string/time, optional] The time of the next scheduled run.

time_zone [string, optional] The time zone of this script.

hidden [boolean, optional] The hidden status of the item.

target_project_id [integer, optional] Target project to which script outputs will be added.

required_resources [dict, optional::]

- **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

- **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

- **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.

instance_type [string, optional] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.

cancel_timeout [integer, optional] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

docker_image_tag [string, optional] The tag of the docker image to pull from DockerHub.

partition_label [string, optional] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.

running_as_id [integer, optional] The ID of the runner of this script.

Returns

civis.response.Response

- **id** [integer] The ID for the script.

- **name** [string] The name of the script.

- **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)

- **created_at** [string/time] The time this script was created.

- **updated_at** [string/time] The time the script was last updated.

- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **state** [string] The status of the script's last run.

- **finished_at** [string/time] The time that the script's last run finished.

- **category** [string] The category of the script.

- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    - **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script

- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.

- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

The variable's name as used within your code.

- **label** [string] The label to present to users when asking them for the value.

- **description** [string] A short sentence or fragment describing this parameter to the end user.

- **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

- **required** [boolean] Whether this param is required.

- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

  - **details** [string] The details link to get more information about the script.

  - **runs** [string] The runs link to get the run information list for this script.
- **schedule** [dict::]

  - **scheduled** [boolean] If the item is scheduled.

  - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  - **scheduled_hours** [list] Hours of the day it is scheduled on.

  - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

  - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

  - **urls** [list] URLs to receive a POST request at job completion

  - **success_email_subject** [string] Custom subject line for success e-mail.

  - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

  - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

  - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

  - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

  - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

  - **success_on** [boolean] If success email notifications are on.

  - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **required_resources** [dict::]

> > > > > – **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.
> > > > >
> > > > > – **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.
> > > > >
> > > > > – **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.
> > > >
> > > > - **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.
> > > > - **source** [string] The body/text of the script.
> > > > - **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.
> > > > - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.
> > > > - **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
> > > > - **running_as_id** [integer] The ID of the runner of this script.

post_r_clone(*id*, ***, *clone_schedule='DEFAULT'*, *clone_triggers='DEFAULT'*, *clone_notifications='DEFAULT'*)

> Clone this R Script
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the script.
> > > **clone_schedule** [boolean, optional] If true, also copy the schedule to the new script.
> > > **clone_triggers** [boolean, optional] If true, also copy the triggers to the new script.
> > > **clone_notifications** [boolean, optional] If true, also copy the notifications to the new script.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **id** [integer] The ID for the script.
> > > > - **name** [string] The name of the script.
> > > > - **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
> > > > - **created_at** [string/time] The time this script was created.
> > > > - **updated_at** [string/time] The time the script was last updated.
> > > > - **author** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > - **state** [string] The status of the script's last run.
> > > > - **finished_at** [string/time] The time that the script's last run finished.
> > > > - **category** [string] The category of the script.
> > > > - **projects** [list::] A list of projects containing the script. - id : integer

> The ID for the project.

> > – **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

> > The variable's name as used within your code.

> > – **label** [string] The label to present to users when asking them for the value.

> > – **description** [string] A short sentence or fragment describing this parameter to the end user.

> > – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

> > – **required** [boolean] Whether this param is required.

> > – **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

> > – **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

> > – **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

> > – **details** [string] The details link to get more information about the script.

> > – **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

> > – **scheduled** [boolean] If the item is scheduled.

- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

- **scheduled_hours** [list] Hours of the day it is scheduled on.

- **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

  - **urls** [list] URLs to receive a POST request at job completion

  - **success_email_subject** [string] Custom subject line for success e-mail.

  - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

  - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

  - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

  - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

  - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

  - **success_on** [boolean] If success email notifications are on.

  - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

- **error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **required_resources** [dict::]

    - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

    - **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

    - **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.

- **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.
- **source** [string] The body/text of the script.
- **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.
- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
- **running_as_id** [integer] The ID of the runner of this script.

post_r_git_checkout(*id*)

Checkout content that the existing git_ref points to and save to the object

> **Parameters**
>> **id** [integer] The ID of the file.
>
> **Returns**
>> *civis.response.Response*
>>> - **content** [string] The file's contents.
>>> - **type** [string] The file's type.
>>> - **size** [integer] The file's size.
>>> - **file_hash** [string] The SHA of the file.

post_r_git_checkout_latest(*id*)

Checkout latest commit on the current branch of a script or workflow

> **Parameters**
>> **id** [integer] The ID of the file.
>
> **Returns**
>> *civis.response.Response*
>>> - **content** [string] The file's contents.
>>> - **type** [string] The file's type.
>>> - **size** [integer] The file's size.
>>> - **file_hash** [string] The SHA of the file.

post_r_git_commits(*id*, *content*, *message*, *file_hash*)

    Commit and push a new version of the file

        Parameters

            id [integer] The ID of the file.

            content [string] The contents to commit to the file.

            message [string] A commit message describing the changes being made.

            file_hash [string] The full SHA of the file being replaced.

        Returns

        civis.response.Response

            • **content** [string] The file's contents.

            • **type** [string] The file's type.

            • **size** [integer] The file's size.

            • **file_hash** [string] The SHA of the file.

post_r_runs(*id*)

    Start a run

        Parameters

            id [integer] The ID of the r.

        Returns

        civis.response.Response

            • **id** [integer] The ID of the run.

            • **r_id** [integer] The ID of the r.

            • **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.

            • **is_cancel_requested** [boolean] True if run cancel requested, else false.

            • **started_at** [string/time] The time the last run started at.

            • **finished_at** [string/time] The time the last run completed.

            • **error** [string] The error, if any, returned by the run.

            • **max_memory_usage** [number/float] If the run has finished, the maximum amount of memory used during the run, in MB.

            • **max_cpu_usage** [number/float] If the run has finished, the maximum amount of cpu used during the run, in millicores.

post_r_runs_outputs(*id*, *run_id*, *object_type*, *object_id*)

    Add an output for a run

        Parameters

            id [integer] The ID of the r script.

            run_id [integer] The ID of the run.

            object_type [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue

            object_id [integer] The ID of the output.

         Returns

         civis.response.Response

            • **object_type** [string] The type of the output. Valid values are File, Table, Report, Project, Credential, or JSONValue

            • **object_id** [integer] The ID of the output.

            • **name** [string] The name of the output.

            • **link** [string] The hypermedia link to the output.

            • value : string

post_run(*id*)

    Run a script

        Parameters

            id [integer] The ID for the script.

        Returns

None Response code 204: success

post_sql(*name*, *sql*, *remote_host_id*, *credential_id*, *, *parent_id='DEFAULT'*, *user_context='DEFAULT'*, *params='DEFAULT'*, *arguments='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*, *next_run_at='DEFAULT'*, *time_zone='DEFAULT'*, *hidden='DEFAULT'*, *target_project_id='DEFAULT'*, *csv_settings='DEFAULT'*, *running_as_id='DEFAULT'*)

Create a SQL script

> **Parameters**
>
> > **name** [string] The name of the script.
> >
> > **sql** [string] The raw SQL query for the script.
> >
> > **remote_host_id** [integer] The remote host ID that this script will connect to.
> >
> > **credential_id** [integer] The credential that this script will use.
> >
> > **parent_id** [integer, optional] The ID of the parent job that will trigger this script
> >
> > **user_context** [string, optional] "runner" or "author", who to execute the script as when run as a template.
> >
> > **params** [list, optional::] A definition of the parameters this script accepts in the arguments field. - name : string
> >
> > > The variable's name as used within your code.
> >
> > > - **label** [string] The label to present to users when asking them for the value.
> > > - **description** [string] A short sentence or fragment describing this parameter to the end user.
> > > - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
> > > - **required** [boolean] Whether this param is required.
> > > - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
> > > - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.
> > > - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*
> >
> > **arguments** [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
> >
> > **schedule** [dict, optional::]
> >
> > > - **scheduled** [boolean] If the item is scheduled.
> > > - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > > - **scheduled_hours** [list] Hours of the day it is scheduled on.
> > > - **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> > > - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
> > > - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
> >
> > **notifications** [dict, optional::]
> >
> > > - **urls** [list] URLs to receive a POST request at job completion
> > > - **success_email_subject** [string] Custom subject line for success e-mail.
> > > - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.
> > > - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

- **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."
- **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.
- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.
- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.
- **success_on** [boolean] If success email notifications are on.
- **failure_on** [boolean] If failure email notifications are on.

next_run_at [string/time, optional] The time of the next scheduled run.

time_zone [string, optional] The time zone of this script.

hidden [boolean, optional] The hidden status of the item.

target_project_id [integer, optional] Target project to which script outputs will be added.

csv_settings [dict, optional::]

- **include_header** [boolean] Whether or not to include headers in the output data. Default: true
- **compression** [string] The type of compression to use, if any, one of "none", "zip", or "gzip". Default: gzip
- **column_delimiter** [string] Which delimiter to use, one of "comma", "tab", or "pipe". Default: comma
- **unquoted** [boolean] Whether or not to quote fields. Default: false
- **force_multifile** [boolean] Whether or not the csv should be split into multiple files. Default: false
- **filename_prefix** [string] A user specified filename prefix for the output file to have. Default: null
- **max_file_size** [integer] The max file size, in MB, created files will be. Only available when force_multifile is true.

running_as_id [integer, optional] The ID of the runner of this script.

Returns

civis.response.Response

- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

  - **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    - **details** [string] The details link to get more information about the script.

    - **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

- **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

  - **urls** [list] URLs to receive a POST request at job completion

  - **success_email_subject** [string] Custom subject line for success e-mail.

  - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

  - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

  - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

  - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

  - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

  - **success_on** [boolean] If success email notifications are on.

  - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.

- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **sql** [string] The raw SQL query for the script.
- **expanded_arguments** [dict] Expanded arguments for use in injecting into different environments.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **code_preview** [string] The code that this script will run with arguments inserted.
- **csv_settings** [dict::]

    – **include_header** [boolean] Whether or not to include headers in the output data. Default: true

    – **compression** [string] The type of compression to use, if any, one of "none", "zip", or "gzip". Default: gzip

    – **column_delimiter** [string] Which delimiter to use, one of "comma", "tab", or "pipe". Default: comma

    – **unquoted** [boolean] Whether or not to quote fields. Default: false

    – **force_multifile** [boolean] Whether or not the csv should be split into multiple files. Default: false

    – **filename_prefix** [string] A user specified filename prefix for the output file to have. Default: null

    – **max_file_size** [integer] The max file size, in MB, created files will be. Only available when force_multifile is true.
- **running_as_id** [integer] The ID of the runner of this script.

post_sql_clone(*id*, ***, *clone_schedule='DEFAULT'*, *clone_triggers='DEFAULT'*, *clone_notifications='DEFAULT'*)

　　Clone this SQL script

　　　　Parameters

　　　　　　id [integer] The ID for the script.

　　　　　　clone_schedule [boolean, optional] If true, also copy the schedule to the new script.

　　　　　　clone_triggers [boolean, optional] If true, also copy the triggers to the new script.

　　　　　　clone_notifications [boolean, optional] If true, also copy the notifications to the new script.

　　　　Returns

　　　　　　civis.response.Response

- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

  The ID for the project.

  - **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

  The variable's name as used within your code.

  - **label** [string] The label to present to users when asking them for the value.

  - **description** [string] A short sentence or fragment describing this parameter to the end user.

  - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

  - **required** [boolean] Whether this param is required.

  - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

  - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

  - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

  - **details** [string] The details link to get more information about the script.

- **runs** [string] The runs link to get the run information list for this script.
- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - id : integer

> > > – **state** : string
> > >
> > > – **created_at** [string/time] The time that the run was queued.
> > >
> > > – **started_at** [string/time] The time that the run started.
> > >
> > > – **finished_at** [string/time] The time that the run completed.
> > >
> > > – **error** [string] The error message for this run, if present.
> >
> > - **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
> > - **hidden** [boolean] The hidden status of the item.
> > - **target_project_id** [integer] Target project to which script outputs will be added.
> > - **archived** [string] The archival status of the requested item(s).
> > - **sql** [string] The raw SQL query for the script.
> > - **expanded_arguments** [dict] Expanded arguments for use in injecting into different environments.
> > - **remote_host_id** [integer] The remote host ID that this script will connect to.
> > - **credential_id** [integer] The credential that this script will use.
> > - **code_preview** [string] The code that this script will run with arguments inserted.
> > - **csv_settings** [dict::]
> >
> > > > – **include_header** [boolean] Whether or not to include headers in the output data. Default: true
> > > >
> > > > – **compression** [string] The type of compression to use, if any, one of "none", "zip", or "gzip". Default: gzip
> > > >
> > > > – **column_delimiter** [string] Which delimiter to use, one of "comma", "tab", or "pipe". Default: comma
> > > >
> > > > – **unquoted** [boolean] Whether or not to quote fields. Default: false
> > > >
> > > > – **force_multifile** [boolean] Whether or not the csv should be split into multiple files. Default: false
> > > >
> > > > – **filename_prefix** [string] A user specified filename prefix for the output file to have. Default: null
> > > >
> > > > – **max_file_size** [integer] The max file size, in MB, created files will be. Only available when force_multifile is true.
> >
> > - **running_as_id** [integer] The ID of the runner of this script.

post_sql_git_checkout(*id*)

> Checkout content that the existing git_ref points to and save to the object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the file.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> >
> > > - **content** [string] The file's contents.
> > > - **type** [string] The file's type.
> > > - **size** [integer] The file's size.
> > > - **file_hash** [string] The SHA of the file.

post_sql_git_checkout_latest(*id*)

> Checkout latest commit on the current branch of a script or workflow

> **Parameters**
>> **id** [integer] The ID of the file.
>
> **Returns**
>> *civis.response.Response*
>>> • **content** [string] The file's contents.
>>> • **type** [string] The file's type.
>>> • **size** [integer] The file's size.
>>> • **file_hash** [string] The SHA of the file.

post_sql_git_commits(*id*, *content*, *message*, *file_hash*)

> Commit and push a new version of the file
>
> **Parameters**
>> **id** [integer] The ID of the file.
>>
>> **content** [string] The contents to commit to the file.
>>
>> **message** [string] A commit message describing the changes being made.
>>
>> **file_hash** [string] The full SHA of the file being replaced.
>
> **Returns**
>> *civis.response.Response*
>>> • **content** [string] The file's contents.
>>> • **type** [string] The file's type.
>>> • **size** [integer] The file's size.
>>> • **file_hash** [string] The SHA of the file.

post_sql_runs(*id*)

> Start a run
>
> **Parameters**
>> **id** [integer] The ID of the sql.
>
> **Returns**
>> *civis.response.Response*
>>> • **id** [integer] The ID of the run.
>>> • **sql_id** [integer] The ID of the sql.
>>> • **state** [string] The state of the run, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
>>> • **is_cancel_requested** [boolean] True if run cancel requested, else false.
>>> • **started_at** [string/time] The time the last run started at.
>>> • **finished_at** [string/time] The time the last run completed.
>>> • **error** [string] The error, if any, returned by the run.
>>> • **output** [list::] A list of the outputs of this script. - output_name : string
>>>
>>>> The name of the output file.
>>>
>>>> – **file_id** [integer] The unique ID of the output file.
>>>>
>>>> – **path** [string] The temporary link to download this output file, valid for 36 hours.
>>>
>>> • **output_cached_on** [string/time] The time that the output was originally exported, if a cache entry was used by the run.

put_containers(*id*, *required_resources*, *docker_image_name*, *, name='DEFAULT'*,
                parent_id='DEFAULT', *user_context='DEFAULT'*, *params='DEFAULT'*,
                arguments='DEFAULT', *schedule='DEFAULT'*, *notifications='DEFAULT'*,
                repo_http_uri='DEFAULT', *repo_ref='DEFAULT'*,
                remote_host_credential_id='DEFAULT', *git_credential_id='DEFAULT'*,
                docker_command='DEFAULT', *docker_image_tag='DEFAULT'*,
                instance_type='DEFAULT', *cancel_timeout='DEFAULT'*, *time_zone='DEFAULT'*,
                partition_label='DEFAULT', *target_project_id='DEFAULT'*, *running_as_id='DEFAULT'*)

      Edit a container

          Parameters

               id [integer] The ID for the script.

               required_resources [dict::]

                      • **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

                      • **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

                      • **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_image_name [string] The name of the docker image to pull from DockerHub.

               name [string, optional] The name of the container.

               parent_id [integer, optional] The ID of the parent job that will trigger this script

               user_context [string, optional] "runner" or "author", who to execute the script as when run as a template.

               params [list, optional::] A definition of the parameters this script accepts in the arguments field. - name : string

                      The variable's name as used within your code.

                      • **label** [string] The label to present to users when asking them for the value.

                      • **description** [string] A short sentence or fragment describing this parameter to the end user.

                      • **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

                      • **required** [boolean] Whether this param is required.

                      • **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

                      • **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

                      • **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

               arguments [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.

               schedule [dict, optional::]

                      • **scheduled** [boolean] If the item is scheduled.

                      • **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

                      • **scheduled_hours** [list] Hours of the day it is scheduled on.

                      • **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

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

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

repo_ref [string, optional] The tag or branch of the github repo to clone into the container.

remote_host_credential_id [integer, optional] The id of the database credentials to pass into the environment of the container.

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

docker_command [string, optional] The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]. Defaults to the Docker image's ENTRYPOINT/CMD.

docker_image_tag [string, optional] The tag of the docker image to pull from DockerHub.

instance_type [string, optional] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.

cancel_timeout [integer, optional] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

time_zone [string, optional] The time zone of this script.

partition_label [string, optional] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.

target_project_id [integer, optional] Target project to which script outputs will be added.

running_as_id [integer, optional] The ID of the runner of this script.

Returns

   civis.response.Response
- **id** [integer] The ID for the script.
- **name** [string] The name of the container.
- **type** [string] The type of the script (e.g Container)
- **created_at** [string/time] The time this script was created.

- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

        The ID for the project.

    - **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

        The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **template_dependents_count** [integer] How many other scripts use this one as a template.

- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template script.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    – **details** [string] The details link to get more information about the script.

    – **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

    – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    – **scheduled_hours** [list] Hours of the day it is scheduled on.

    – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    – **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    – **urls** [list] URLs to receive a POST request at job completion

    – **success_email_subject** [string] Custom subject line for success e-mail.

    – **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    – **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    – **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    – **success_on** [boolean] If success email notifications are on.

    – **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.

- **required_resources** [dict::]

  - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

  - **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

  - **disk_space** [number/float] The amount of disk space, in GB, to allocate for the container. This space will be used to hold the git repo configured for the container and anything your container writes to /tmp or /data. Fractional values (e.g. 0.25) are supported.

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

- **repo_ref** [string] The tag or branch of the github repo to clone into the container.

- **remote_host_credential_id** [integer] The id of the database credentials to pass into the environment of the container.

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

- **docker_command** [string] The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]. Defaults to the Docker image's ENTRYPOINT/CMD.

- **docker_image_name** [string] The name of the docker image to pull from DockerHub.

- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.

- **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.

- **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

- **time_zone** [string] The time zone of this script.

- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **target_project_id** [integer] Target project to which script outputs will be added.
- **running_as_id** [integer] The ID of the runner of 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

civis.response.Response

- **id** [integer] The ID for the script.
- **name** [string] The name of the container.
- **type** [string] The type of the script (e.g Container)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

  The ID for the project.

  – **name** [string] The name of the project.
- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

  The variable's name as used within your code.

  – **label** [string] The label to present to users when asking them for the value.

  – **description** [string] A short sentence or fragment describing this parameter to the end user.

  – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

  – **required** [boolean] Whether this param is required.

- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template script.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    - **details** [string] The details link to get more information about the script.

    - **runs** [string] The runs link to get the run information list for this script.
- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

- **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

- **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **required_resources** [dict::]

  - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

  - **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

  - **disk_space** [number/float] The amount of disk space, in GB, to allocate for the container. This space will be used to hold the git repo configured for the container and anything your container writes to /tmp or /data. Fractional values (e.g. 0.25) are supported.

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

- **repo_ref** [string] The tag or branch of the github repo to clone into the container.

- **remote_host_credential_id** [integer] The id of the database credentials to pass into the environment of the container.

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

- **docker_command** [string] The command to run on the container. Will be run via sh as: ["sh", "-c", dockerCommand]. Defaults to the Docker image's ENTRYPOINT/CMD.

- **docker_image_name** [string] The name of the docker image to pull from DockerHub.

- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.

- **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.

- **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.
- **last_run** [dict::]

    – id : integer

    – state : string

    – **created_at** [string/time] The time that the run was queued.

    – **started_at** [string/time] The time that the run started.

    – **finished_at** [string/time] The time that the run completed.

    – **error** [string] The error message for this run, if present.
- **time_zone** [string] The time zone of this script.
- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **target_project_id** [integer] Target project to which script outputs will be added.
- **running_as_id** [integer] The ID of the runner of this script.

put_containers_projects(*id*, *project_id*)

Add a Container Script to a project

> **Parameters**
>> **id** [integer] The ID of the Container Script.
>> **project_id** [integer] The ID of the project.
>
> **Returns**
>> **None** Response code 204: success

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

Set the permissions groups has on this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>> **group_ids** [list] An array of one or more group IDs.
>> **permission_level** [string] Options are: "read", "write", or "manage".
>> **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
>> **send_shared_email** [boolean, optional] Send email to the recipients of a share.
>
> **Returns**
>> *civis.response.Response*
>>> - **readers** [dict::]
>>>
>>>   – **users** [list::]
>>>
>>>   ∗ id : integer
>>>
>>>   ∗ name : string
>>>
>>>   – **groups** [list::]
>>>
>>>   ∗ id : integer
>>>
>>>   ∗ name : string

- **writers** [dict::]

    – **users** [list::]

        ∗ id : integer

        ∗ name : string

    – **groups** [list::]

        ∗ id : integer

        ∗ name : string

- **owners** [dict::]

    – **users** [list::]

        ∗ id : integer

        ∗ name : string

    – **groups** [list::]

        ∗ id : integer

        ∗ name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

 Set the permissions users have on this object

  Parameters

   id [integer] The ID of the resource that is shared.

   user_ids [list] An array of one or more user IDs.

   permission_level [string] Options are: "read", "write", or "manage".

   share_email_body [string, optional] Custom body text for e-mail sent on a share.

   send_shared_email [boolean, optional] Send email to the recipients of a share.

  Returns

  [*civis.response.Response*](#)

- **readers** [dict::]

    – **users** [list::]

        ∗ id : integer

        ∗ name : string

    – **groups** [list::]

        ∗ id : integer

        ∗ name : string

- **writers** [dict::]

    – **users** [list::]

        ∗ id : integer

        ∗ name : string

    – **groups** [list::]

        ∗ id : integer

* name : string
- **owners** [dict::]

  – **users** [list::]

    * id : integer

    * name : string

  – **groups** [list::]

    * id : integer

    * name : string
- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_containers_transfer(*id*, *user_id*, *include_dependencies*, ***, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

Transfer ownership of this object to another user

> **Parameters**
>
>> **id** [integer] The ID of the resource that is shared.
>>
>> **user_id** [integer] ID of target user
>>
>> **include_dependencies** [boolean] Whether or not to give manage permissions on all dependencies
>>
>> **email_body** [string, optional] Custom body text for e-mail sent on transfer.
>>
>> **send_email** [boolean, optional] Send email to the target user of the transfer?
>
> **Returns**
>
> [*civis.response.Response*](#)
>
>> - **dependencies** [list::] Dependent objects for this object - object_type : string
>>
>>> Dependent object type
>>
>>   – **fco_type** [string] Human readable dependent object type
>>
>>   – **id** [integer] Dependent object ID
>>
>>   – **name** [string] Dependent object name, or nil if the requesting user cannot read this object
>>
>>   – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
>>
>>   – **shared** [boolean] Whether dependent object was successfully shared with target user

put_custom(*id*, ***, *name='DEFAULT'*, *parent_id='DEFAULT'*, *arguments='DEFAULT'*, *remote_host_id='DEFAULT'*, *credential_id='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*, *time_zone='DEFAULT'*, *target_project_id='DEFAULT'*, *required_resources='DEFAULT'*, *partition_label='DEFAULT'*, *running_as_id='DEFAULT'*)

Replace all attributes of this Custom Script

> **Parameters**
>
>> **id** [integer] The ID for the script.
>>
>> **name** [string, optional] The name of the script.
>>
>> **parent_id** [integer, optional] The ID of the parent job that will trigger this script

---

arguments [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.

remote_host_id [integer, optional] The remote host ID that this script will connect to.

credential_id [integer, optional] The credential that this script will use.

schedule [dict, optional::]
- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

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

time_zone [string, optional] The time zone of this script.

target_project_id [integer, optional] Target project to which script outputs will be added.

required_resources [dict, optional::]
- **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares.
- **memory** [integer] The amount of RAM to allocate for the container (in MB).
- **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.

partition_label [string, optional] The partition label used to run this object. Only applicable for jobs using Docker.Not generally available. Beware this attribute may be removed in the future.

running_as_id [integer, optional] The ID of the runner of this script.

Returns

civis.response.Response
- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g Custom)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.

- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** : string
- **projects** [list::] A list of projects containing the script. - id : integer

        The ID for the project.

    - **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

        The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template script.
- **ui_report_url** [integer] The url of the custom HTML.
- **ui_report_id** [integer] The id of the report with the custom HTML.

- **ui_report_provide_api_key** [boolean] Whether the ui report requests an API Key from the report viewer.
- **template_script_name** [string] The name of the template script.
- **template_note** [string] The template's note.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **code_preview** [string] The code that this script will run with arguments inserted.
- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **target_project_id** [integer] Target project to which script outputs will be added.
- **last_successful_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **required_resources** [dict::]

    - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares.

    - **memory** [integer] The amount of RAM to allocate for the container (in MB).

    - **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.
- **partition_label** [string] The partition label used to run this object. Only applicable for jobs using Docker.Not generally available. Beware this attribute may be removed in the future.
- **running_as_id** [integer] The ID of the runner of this script.

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**
> >
> > > *civis.response.Response*
> > >
> > > - **id** [integer] The ID for the script.
> > > - **name** [string] The name of the script.
> > > - **type** [string] The type of the script (e.g Custom)

- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- category : string
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    - **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template script.

- **ui_report_url** [integer] The url of the custom HTML.
- **ui_report_id** [integer] The id of the report with the custom HTML.
- **ui_report_provide_api_key** [boolean] Whether the ui report requests an API Key from the report viewer.
- **template_script_name** [string] The name of the template script.
- **template_note** [string] The template's note.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **code_preview** [string] The code that this script will run with arguments inserted.
- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **archived** [string] The archival status of the requested item(s).
- **target_project_id** [integer] Target project to which script outputs will be added.
- **last_successful_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.
- **required_resources** [dict::]

  - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares.

  - **memory** [integer] The amount of RAM to allocate for the container (in MB).

  - **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.
- **partition_label** [string] The partition label used to run this object. Only applicable for jobs using Docker.Not generally available. Beware this attribute may be removed in the future.
- **running_as_id** [integer] The ID of the runner of this script.

put_custom_projects(*id*, *project_id*)

Add a Custom Script to a project

> **Parameters**
>> **id** [integer] The ID of the Custom Script.
>> **project_id** [integer] The ID of the project.
>
> **Returns**
>> **None** Response code 204: success

---

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

    Set the permissions groups has on this object

        Parameters

            id [integer] The ID of the resource that is shared.

            group_ids [list] An array of one or more group IDs.

            permission_level [string] Options are: "read", "write", or "manage".

            share_email_body [string, optional] Custom body text for e-mail sent on a share.

            send_shared_email [boolean, optional] Send email to the recipients of a share.

        Returns

        [*civis.response.Response*](#)

- **readers** [dict::]

  - **users** [list::]

    * id : integer

    * name : string

  - **groups** [list::]

    * id : integer

    * name : string

- **writers** [dict::]

  - **users** [list::]

    * id : integer

    * name : string

  - **groups** [list::]

    * id : integer

    * name : string

- **owners** [dict::]

  - **users** [list::]

    * id : integer

    * name : string

  - **groups** [list::]

    * id : integer

    * name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

    Set the permissions users have on this object

        Parameters

            id [integer] The ID of the resource that is shared.

            user_ids [list] An array of one or more user IDs.

            permission_level [string] Options are: "read", "write", or "manage".

            share_email_body [string, optional] Custom body text for e-mail sent on a share.

> **send_shared_email** [boolean, optional] Send email to the recipients of a share.

Returns

civis.response.Response

- **readers** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **writers** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **owners** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_custom_transfer(*id*, *user_id*, *include_dependencies*, ***, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

> Transfer ownership of this object to another user

Parameters

id [integer] The ID of the resource that is shared.

user_id [integer] ID of target user

include_dependencies [boolean] Whether or not to give manage permissions on all dependencies

email_body [string, optional] Custom body text for e-mail sent on transfer.

send_email [boolean, optional] Send email to the target user of the transfer?

Returns

civis.response.Response

- **dependencies** [list::] Dependent objects for this object - object_type : string

    Dependent object type

> > > > – **fco_type** [string] Human readable dependent object type
> > > >
> > > > – **id** [integer] Dependent object ID
> > > >
> > > > – **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > > >
> > > > – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > > >
> > > > – **shared** [boolean] Whether dependent object was successfully shared with target user

put_javascript(*id*, *name*, *source*, *remote_host_id*, *credential_id*, ***, *parent_id='DEFAULT'*, *user_context='DEFAULT'*, *params='DEFAULT'*, *arguments='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*, *next_run_at='DEFAULT'*, *time_zone='DEFAULT'*, *target_project_id='DEFAULT'*, *running_as_id='DEFAULT'*)

> Replace all attributes of this JavaScript Script
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the script.
> > >
> > > **name** [string] The name of the script.
> > >
> > > **source** [string] The body/text of the script.
> > >
> > > **remote_host_id** [integer] The remote host ID that this script will connect to.
> > >
> > > **credential_id** [integer] The credential that this script will use.
> > >
> > > **parent_id** [integer, optional] The ID of the parent job that will trigger this script
> > >
> > > **user_context** [string, optional] "runner" or "author", who to execute the script as when run as a template.
> > >
> > > **params** [list, optional::] A definition of the parameters this script accepts in the arguments field. - name : string
> > >
> > > > The variable's name as used within your code.
> > > >
> > > > - **label** [string] The label to present to users when asking them for the value.
> > > > - **description** [string] A short sentence or fragment describing this parameter to the end user.
> > > > - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
> > > > - **required** [boolean] Whether this param is required.
> > > > - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
> > > > - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.
> > > > - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*
> > >
> > > **arguments** [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
> > >
> > > **schedule** [dict, optional::]
> > >
> > > > - **scheduled** [boolean] If the item is scheduled.
> > > > - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > > > - **scheduled_hours** [list] Hours of the day it is scheduled on.
> > > > - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

notifications [dict, optional::]

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

next_run_at [string/time, optional] The time of the next scheduled run.

time_zone [string, optional] The time zone of this script.

target_project_id [integer, optional] Target project to which script outputs will be added.

running_as_id [integer, optional] The ID of the runner of this script.

Returns

[*civis.response.Response*](#)

- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

  - **name** [string] The name of the project.
- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.

- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    – **label** [string] The label to present to users when asking them for the value.

    – **description** [string] A short sentence or fragment describing this parameter to the end user.

    – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    – **required** [boolean] Whether this param is required.

    – **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    – **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    – **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    – **details** [string] The details link to get more information about the script.

    – **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

    – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    – **scheduled_hours** [list] Hours of the day it is scheduled on.

    – **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).

- **source** [string] The body/text of the script.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **running_as_id** [integer] The ID of the runner of 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**
>>> *civis.response.Response*

- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

  – **name** [string] The name of the project.
- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

  – **label** [string] The label to present to users when asking them for the value.

  – **description** [string] A short sentence or fragment describing this parameter to the end user.

  – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

  – **required** [boolean] Whether this param is required.

- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

  - **details** [string] The details link to get more information about the script.

  - **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

  - **scheduled** [boolean] If the item is scheduled.

  - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  - **scheduled_hours** [list] Hours of the day it is scheduled on.

  - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

  - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

  - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

  - **urls** [list] URLs to receive a POST request at job completion

  - **success_email_subject** [string] Custom subject line for success e-mail.

  - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

- **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

- **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **source** [string] The body/text of the script.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **running_as_id** [integer] The ID of the runner of this script.

put_javascript_git(*id*, ***, *git_ref='DEFAULT'*, *git_branch='DEFAULT'*, *git_path='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref_type='DEFAULT'*, *pull_from_git='DEFAULT'*)

   Attach an item to a file in a git repo

   Parameters

   id [integer] The ID of the file.

   git_ref [string, optional] A git reference specifying an unambiguous version of the file. Can be a branch name, or the full or shortened SHA of a commit.

   git_branch [string, optional] The git branch that the file is on.

   git_path [string, optional] The path of the file in the repository.

> > **git_repo_url** [string, optional] The URL of the git repository.
> > **git_ref_type** [string, optional] Specifies if the file is versioned by branch or tag.
> > **pull_from_git** [boolean, optional] Automatically pull latest commit from git. Only
> > > works for scripts.
>
> > **Returns**
> > > [*civis.response.Response*](#)
> > >
> > > > • **git_ref** [string] A git reference specifying an unambiguous version of the
> > > >   file. Can be a branch name, tag or the full or shortened SHA of a
> > > >   commit.
> > > > • **git_branch** [string] The git branch that the file is on.
> > > > • **git_path** [string] The path of the file in the repository.
> > > > • **git_repo** [dict::]
> > > >
> > > > > – **id** [integer] The ID for this git repository.
> > > > >
> > > > > – **repo_url** [string] The URL for this git repository.
> > > > >
> > > > > – created_at : string/time
> > > > >
> > > > > – updated_at : string/time
> > > >
> > > > • **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
> > > > • **pull_from_git** [boolean] Automatically pull latest commit from git. Only
> > > >   works for scripts and workflows (assuming you have the feature en-
> > > >   abled)

put_javascript_projects(*id*, *project_id*)
> Add a JavaScript Script to a project
> > **Parameters**
> > > **id** [integer] The ID of the JavaScript Script.
> > > **project_id** [integer] The ID of the project.
> > **Returns**
> > > **None** Response code 204: success

put_javascript_shares_groups(*id*, *group_ids*, *permission_level*, ***, *share_email_body='DEFAULT'*,
> > *send_shared_email='DEFAULT'*)
> Set the permissions groups has on this object
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **group_ids** [list] An array of one or more group IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> > **Returns**
> > > [*civis.response.Response*](#)
> > >
> > > > • **readers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **writers** [dict::]
> > > >
> > > > > – **users** [list::]

           ∗ id : integer

           ∗ name : string

        – **groups** [list::]

           ∗ id : integer

           ∗ name : string

      • **owners** [dict::]

        – **users** [list::]

           ∗ id : integer

           ∗ name : string

        – **groups** [list::]

           ∗ id : integer

           ∗ name : string

      • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

      • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

  Set the permissions users have on this object

    Parameters

      id [integer] The ID of the resource that is shared.

      user_ids [list] An array of one or more user IDs.

      permission_level [string] Options are: "read", "write", or "manage".

      share_email_body [string, optional] Custom body text for e-mail sent on a share.

      send_shared_email [boolean, optional] Send email to the recipients of a share.

    Returns

    [`civis.response.Response`](#)

      • **readers** [dict::]

        – **users** [list::]

           ∗ id : integer

           ∗ name : string

        – **groups** [list::]

           ∗ id : integer

           ∗ name : string

      • **writers** [dict::]

        – **users** [list::]

           ∗ id : integer

           ∗ name : string

        – **groups** [list::]

           ∗ id : integer

           ∗ name : string

      • **owners** [dict::]

> > > > > – **users** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string

> > > > > – **groups** [list::]

> > > > > > ∗ id : integer

> > > > > > ∗ name : string
> > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_javascript_transfer(*id*, *user_id*, *include_dependencies*, ***, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

> Transfer ownership of this object to another user
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_id** [integer] ID of target user
> > > **include_dependencies** [boolean] Whether or not to give manage permissions on all dependencies
> > > **email_body** [string, optional] Custom body text for e-mail sent on transfer.
> > > **send_email** [boolean, optional] Send email to the target user of the transfer?
> > **Returns**
> > > [*civis.response.Response*](#)
> > > > • **dependencies** [list::] Dependent objects for this object - object_type : string

> > > > > Dependent object type

> > > > > – **fco_type** [string] Human readable dependent object type

> > > > > – **id** [integer] Dependent object ID

> > > > > – **name** [string] Dependent object name, or nil if the requesting user cannot read this object

> > > > > – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user

> > > > > – **shared** [boolean] Whether dependent object was successfully shared with target user

put_python3(*id*, *name*, *source*, ***, *parent_id='DEFAULT'*, *user_context='DEFAULT'*, *params='DEFAULT'*, *arguments='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*, *next_run_at='DEFAULT'*, *time_zone='DEFAULT'*, *target_project_id='DEFAULT'*, *required_resources='DEFAULT'*, *instance_type='DEFAULT'*, *cancel_timeout='DEFAULT'*, *docker_image_tag='DEFAULT'*, *partition_label='DEFAULT'*, *running_as_id='DEFAULT'*)

> Replace all attributes of this Python Script
> > **Parameters**
> > > **id** [integer] The ID for the script.
> > > **name** [string] The name of the script.
> > > **source** [string] The body/text of the script.
> > > **parent_id** [integer, optional] The ID of the parent job that will trigger this script
> > > **user_context** [string, optional] "runner" or "author", who to execute the script as when run as a template.

params [list, optional::] A definition of the parameters this script accepts in the arguments field. - name : string

> The variable's name as used within your code.

- **label** [string] The label to present to users when asking them for the value.
- **description** [string] A short sentence or fragment describing this parameter to the end user.
- **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
- **required** [boolean] Whether this param is required.
- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.
- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

arguments [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.

schedule [dict, optional::]
- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

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

next_run_at [string/time, optional] The time of the next scheduled run.

time_zone [string, optional] The time zone of this script.

target_project_id [integer, optional] Target project to which script outputs will be added.

required_resources [dict, optional::]

- **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.
- **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.
- **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.

instance_type [string, optional] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.

cancel_timeout [integer, optional] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

docker_image_tag [string, optional] The tag of the docker image to pull from DockerHub.

partition_label [string, optional] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.

running_as_id [integer, optional] The ID of the runner of this script.

Returns

   civis.response.Response

- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

  - **name** [string] The name of the project.
- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

  - **label** [string] The label to present to users when asking them for the value.

- **description** [string] A short sentence or fragment describing this parameter to the end user.

- **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

- **required** [boolean] Whether this param is required.

- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

  - **details** [string] The details link to get more information about the script.

  - **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

  - **scheduled** [boolean] If the item is scheduled.

  - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  - **scheduled_hours** [list] Hours of the day it is scheduled on.

  - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

  - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

  - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

  - **urls** [list] URLs to receive a POST request at job completion

- **success_email_subject** [string] Custom subject line for success e-mail.

- **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

- **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

- **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

- **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **required_resources** [dict::]

  - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

  - **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

- **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.

- **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.
- **source** [string] The body/text of the script.
- **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.
- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
- **running_as_id** [integer] The ID of the runner of 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**
>> *civis.response.Response*
>>
>> - **id** [integer] The ID for the script.
>> - **name** [string] The name of the script.
>> - **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
>> - **created_at** [string/time] The time this script was created.
>> - **updated_at** [string/time] The time the script was last updated.
>> - **author** [dict::]
>>
>>> - **id** [integer] The ID of this user.
>>>
>>> - **name** [string] This user's name.
>>>
>>> - **username** [string] This user's username.
>>>
>>> - **initials** [string] This user's initials.
>>>
>>> - **online** [boolean] Whether this user is online.
>>
>> - **state** [string] The status of the script's last run.
>> - **finished_at** [string/time] The time that the script's last run finished.
>> - **category** [string] The category of the script.
>> - **projects** [list::] A list of projects containing the script. - id : integer
>>
>>> The ID for the project.
>>
>>> - **name** [string] The name of the project.
>>
>> - **parent_id** [integer] The ID of the parent job that will trigger this script
>> - **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
>> - **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

The variable's name as used within your code.

- **label** [string] The label to present to users when asking them for the value.

- **description** [string] A short sentence or fragment describing this parameter to the end user.

- **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

- **required** [boolean] Whether this param is required.

- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

  - **details** [string] The details link to get more information about the script.

  - **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

  - **scheduled** [boolean] If the item is scheduled.

  - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  - **scheduled_hours** [list] Hours of the day it is scheduled on.

  - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

  - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **required_resources** [dict::]

> > > – **cpu** [integer] The number of CPU shares to allocate for the
> > >   container. Each core has 1000 shares. Must be at least 2
> > >   shares.
> > >
> > > – **memory** [integer] The amount of RAM to allocate for the
> > >   container (in MB). Must be at least 4 MB.
> > >
> > > – **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 any-
> > >   thing your container writes to /tmp or /data. Fractional
> > >   values (e.g. 0.25) are supported.
> >
> > - **instance_type** [string] The EC2 instance type to deploy to. Only available
> >   for jobs running on kubernetes.
> > - **source** [string] The body/text of the script.
> > - **cancel_timeout** [integer] The amount of time (in seconds) to wait before
> >   forcibly terminating the script. When the script is cancelled, it is first
> >   sent a TERM signal. If the script is still running after the timeout, it
> >   is sent a KILL signal. Defaults to 0.
> > - **docker_image_tag** [string] The tag of the docker image to pull from
> >   DockerHub.
> > - **partition_label** [string] The partition label used to run this object. Not
> >   generally available. Beware this attribute may be removed in the
> >   future.
> > - **running_as_id** [integer] The ID of the runner of this script.

put_python3_git(*id*, ***, *git_ref='DEFAULT'*, *git_branch='DEFAULT'*, *git_path='DEFAULT'*,
git_repo_url='DEFAULT', *git_ref_type='DEFAULT'*, *pull_from_git='DEFAULT'*)

> Attach an item to a file in a git repo
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the file.
> > >
> > > **git_ref** [string, optional] A git reference specifying an unambiguous version of the file.
> > > Can be a branch name, or the full or shortened SHA of a commit.
> > >
> > > **git_branch** [string, optional] The git branch that the file is on.
> > >
> > > **git_path** [string, optional] The path of the file in the repository.
> > >
> > > **git_repo_url** [string, optional] The URL of the git repository.
> > >
> > > **git_ref_type** [string, optional] Specifies if the file is versioned by branch or tag.
> > >
> > > **pull_from_git** [boolean, optional] Automatically pull latest commit from git. Only
> > > works for scripts.
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > - **git_ref** [string] A git reference specifying an unambiguous version of the
> > >   file. Can be a branch name, tag or the full or shortened SHA of a
> > >   commit.
> > > - **git_branch** [string] The git branch that the file is on.
> > > - **git_path** [string] The path of the file in the repository.
> > > - **git_repo** [dict::]
> > >
> > > > – **id** [integer] The ID for this git repository.
> > > >
> > > > – **repo_url** [string] The URL for this git repository.
> > > >
> > > > – created_at : string/time
> > > >
> > > > – updated_at : string/time
> > >
> > > - **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
> > > - **pull_from_git** [boolean] Automatically pull latest commit from git. Only

works for scripts and workflows (assuming you have the feature enabled)

put_python3_projects(*id*, *project_id*)

>Add a Python Script to a project

>>**Parameters**

>>>**id** [integer] The ID of the Python Script.

>>>**project_id** [integer] The ID of the project.

>>**Returns**

>>>**None** Response code 204: success

put_python3_shares_groups(*id*, *group_ids*, *permission_level*, *, *share_email_body='DEFAULT'*, *send_shared_email='DEFAULT'*)

>Set the permissions groups has on this object

>>**Parameters**

>>>**id** [integer] The ID of the resource that is shared.

>>>**group_ids** [list] An array of one or more group IDs.

>>>**permission_level** [string] Options are: "read", "write", or "manage".

>>>**share_email_body** [string, optional] Custom body text for e-mail sent on a share.

>>>**send_shared_email** [boolean, optional] Send email to the recipients of a share.

>>**Returns**

>>>[*civis.response.Response*](#)

>>>>• **readers** [dict::]

>>>>>– **users** [list::]

>>>>>>∗ id : integer

>>>>>>∗ name : string

>>>>>– **groups** [list::]

>>>>>>∗ id : integer

>>>>>>∗ name : string

>>>>• **writers** [dict::]

>>>>>– **users** [list::]

>>>>>>∗ id : integer

>>>>>>∗ name : string

>>>>>– **groups** [list::]

>>>>>>∗ id : integer

>>>>>>∗ name : string

>>>>• **owners** [dict::]

>>>>>– **users** [list::]

>>>>>>∗ id : integer

>>>>>>∗ name : string

>>>>>– **groups** [list::]

>>>>>>∗ id : integer

>>>>>>∗ name : string

>>>>• **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_python3_shares_users(*id*, *user_ids*, *permission_level*, ***, *share_email_body='DEFAULT'*, *send_shared_email='DEFAULT'*)

    Set the permissions users have on this object

        Parameters

            id [integer] The ID of the resource that is shared.

            user_ids [list] An array of one or more user IDs.

            permission_level [string] Options are: "read", "write", or "manage".

            share_email_body [string, optional] Custom body text for e-mail sent on a share.

            send_shared_email [boolean, optional] Send email to the recipients of a share.

        Returns

        civis.response.Response

- **readers** [dict::]
    - **users** [list::]
        * id : integer
        * name : string
    - **groups** [list::]
        * id : integer
        * name : string
- **writers** [dict::]
    - **users** [list::]
        * id : integer
        * name : string
    - **groups** [list::]
        * id : integer
        * name : string
- **owners** [dict::]
    - **users** [list::]
        * id : integer
        * name : string
    - **groups** [list::]
        * id : integer
        * name : string
- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_python3_transfer(*id*, *user_id*, *include_dependencies*, ***, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

    Transfer ownership of this object to another user

        Parameters

            id [integer] The ID of the resource that is shared.

            user_id [integer] ID of target user

> include_dependencies [boolean] Whether or not to give manage permissions on all
> dependencies
>
> email_body [string, optional] Custom body text for e-mail sent on transfer.
>
> send_email [boolean, optional] Send email to the target user of the transfer?
>
> **Returns**
>
> > *civis.response.Response*
>
> > > • **dependencies** [list::] Dependent objects for this object - object_type :
> > > string
> > >
> > > > Dependent object type
> > >
> > > > – **fco_type** [string] Human readable dependent object type
> > > >
> > > > – **id** [integer] Dependent object ID
> > > >
> > > > – **name** [string] Dependent object name, or nil if the requesting
> > > > user cannot read this object
> > > >
> > > > – **permission_level** [string] Permission level of target user (not
> > > > user's groups) for dependent object, or null if no target
> > > > user
> > > >
> > > > – **shared** [boolean] Whether dependent object was successfully
> > > > shared with target user

put_r(*id*, *name*, *source*, ***, *parent_id='DEFAULT'*, *user_context='DEFAULT'*, *params='DEFAULT'*,
arguments='DEFAULT', *schedule='DEFAULT'*, *notifications='DEFAULT'*, *next_run_at='DEFAULT'*,
time_zone='DEFAULT', *target_project_id='DEFAULT'*, *required_resources='DEFAULT'*,
instance_type='DEFAULT', *cancel_timeout='DEFAULT'*, *docker_image_tag='DEFAULT'*,
partition_label='DEFAULT', *running_as_id='DEFAULT'*)

> Replace all attributes of this R Script
>
> **Parameters**
>
> > **id** [integer] The ID for the script.
> >
> > **name** [string] The name of the script.
> >
> > **source** [string] The body/text of the script.
> >
> > **parent_id** [integer, optional] The ID of the parent job that will trigger this script
> >
> > **user_context** [string, optional] "runner" or "author", who to execute the script as when
> > run as a template.
> >
> > **params** [list, optional::] A definition of the parameters this script accepts in the argu-
> > ments field. - name : string
> >
> > > The variable's name as used within your code.
> >
> > > • **label** [string] The label to present to users when asking them for the value.
> > > • **description** [string] A short sentence or fragment describing this param-
> > > eter to the end user.
> > > • **type** [string] The type of parameter. Valid options: string,
> > > multi_line_string, integer, float, bool, file, table, database, cre-
> > > dential_aws, credential_redshift, or credential_custom
> > > • **required** [boolean] Whether this param is required.
> > > • **value** [string] The value you would like to set this param to. Setting this
> > > value makes this parameter a fixed param.
> > > • **default** [string] If an argument for this parameter is not defined, it will
> > > use this default value. Use true, True, t, y, yes, or 1 for true bool's
> > > or false, False, f, n, no, or 0 for false bool's. Cannot be used for
> > > parameters that are required or a credential type.
> > > • **allowed_values** [list] The possible values this parameter can take, effec-
> > > tively making this an enumerable parameter. Allowed values is an

> > array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

arguments [dict, optional] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.

schedule [dict, optional::]

- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

notifications [dict, optional::]

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

next_run_at [string/time, optional] The time of the next scheduled run.

time_zone [string, optional] The time zone of this script.

target_project_id [integer, optional] Target project to which script outputs will be added.

required_resources [dict, optional::]

- **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.
- **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.
- **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.

instance_type [string, optional] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.

cancel_timeout [integer, optional] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

docker_image_tag [string, optional] The tag of the docker image to pull from DockerHub.

partition_label [string, optional] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.

> > > **running_as_id** [integer, optional] The ID of the runner of this script.
> > **Returns**
> > *civis.response.Response*
>
> - **id** [integer] The ID for the script.
> - **name** [string] The name of the script.
> - **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
> - **created_at** [string/time] The time this script was created.
> - **updated_at** [string/time] The time the script was last updated.
> - **author** [dict::]
>
> > - **id** [integer] The ID of this user.
> >
> > - **name** [string] This user's name.
> >
> > - **username** [string] This user's username.
> >
> > - **initials** [string] This user's initials.
> >
> > - **online** [boolean] Whether this user is online.
>
> - **state** [string] The status of the script's last run.
> - **finished_at** [string/time] The time that the script's last run finished.
> - **category** [string] The category of the script.
> - **projects** [list::] A list of projects containing the script. - id : integer
>
> > The ID for the project.
> >
> > - **name** [string] The name of the project.
>
> - **parent_id** [integer] The ID of the parent job that will trigger this script
> - **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
> - **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string
>
> > The variable's name as used within your code.
> >
> > - **label** [string] The label to present to users when asking them for the value.
> >
> > - **description** [string] A short sentence or fragment describing this parameter to the end user.
> >
> > - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
> >
> > - **required** [boolean] Whether this param is required.
> >
> > - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
> >
> > - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.
> >
> > - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter.

Allowed values is an array of hashes of the following
format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script.
  Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this
  script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this
  one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    - **details** [string] The details link to get more information about
      the script.

    - **runs** [string] The runs link to get the run information list for
      this script.

- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric
      value starting at 0 for Sunday. Mutually exclusive with
      scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled
      on.

    - **scheduled_runs_per_hour** [integer] Alternative to sched-
      uled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is
      scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for suc-
      cess e-mail.

    - **success_email_body** [string] Custom body text for success
      e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail
      when the job completes successfully.

    - **success_email_from_name** [string] Name from which suc-
      cess emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to suc-
      cess emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail
      when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will
      be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.
- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **required_resources** [dict::]

    - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

    - **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

    - **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.
- **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.
- **source** [string] The body/text of the script.
- **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.
- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
- **running_as_id** [integer] The ID of the runner of 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**
> >
> > > *civis.response.Response*
> > >
> > > > • **id** [integer] The ID for the script.
> > > > • **name** [string] The name of the script.
> > > > • **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
> > > > • **created_at** [string/time] The time this script was created.
> > > > • **updated_at** [string/time] The time the script was last updated.
> > > > • **author** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > • **state** [string] The status of the script's last run.
> > > > • **finished_at** [string/time] The time that the script's last run finished.
> > > > • **category** [string] The category of the script.
> > > > • **projects** [list::] A list of projects containing the script. - id : integer
> > > >
> > > > > The ID for the project.
> > > >
> > > > > – **name** [string] The name of the project.
> > > >
> > > > • **parent_id** [integer] The ID of the parent job that will trigger this script
> > > > • **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
> > > > • **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string
> > > >
> > > > > The variable's name as used within your code.
> > > >
> > > > > – **label** [string] The label to present to users when asking them for the value.
> > > > >
> > > > > – **description** [string] A short sentence or fragment describing this parameter to the end user.
> > > > >
> > > > > – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
> > > > >
> > > > > – **required** [boolean] Whether this param is required.
> > > > >
> > > > > – **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
> > > > >
> > > > > – **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0

> > > for false bool's. Cannot be used for parameters that are
> > > required or a credential type.

> > – **allowed_values** [list] The possible values this parameter can
> > take, effectively making this an enumerable parameter.
> > Allowed values is an array of hashes of the following
> > format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script.
  Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this
  script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this
  one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

> > – **details** [string] The details link to get more information about
> > the script.

> > – **runs** [string] The runs link to get the run information list for
> > this script.

- **schedule** [dict::]

> > – **scheduled** [boolean] If the item is scheduled.

> > – **scheduled_days** [list] Days of the week, based on numeric
> > value starting at 0 for Sunday. Mutually exclusive with
> > scheduledDaysOfMonth

> > – **scheduled_hours** [list] Hours of the day it is scheduled on.

> > – **scheduled_minutes** [list] Minutes of the day it is scheduled
> > on.

> > – **scheduled_runs_per_hour** [integer] Alternative to sched-
> > uled minutes, number of times to run per hour.

> > – **scheduled_days_of_month** [list] Days of the month it is
> > scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

> > – **urls** [list] URLs to receive a POST request at job completion

> > – **success_email_subject** [string] Custom subject line for suc-
> > cess e-mail.

> > – **success_email_body** [string] Custom body text for success
> > e-mail, written in Markdown.

> > – **success_email_addresses** [list] Addresses to notify by e-mail
> > when the job completes successfully.

> > – **success_email_from_name** [string] Name from which suc-
> > cess emails are sent; defaults to "Civis."

> > – **success_email_reply_to** [string] Address for replies to suc-
> > cess emails; defaults to the author of the job.

- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on.

- **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

  - id : integer

  - state : string

  - **created_at** [string/time] The time that the run was queued.

  - **started_at** [string/time] The time that the run started.

  - **finished_at** [string/time] The time that the run completed.

  - **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **required_resources** [dict::]

  - **cpu** [integer] The number of CPU shares to allocate for the container. Each core has 1000 shares. Must be at least 2 shares.

  - **memory** [integer] The amount of RAM to allocate for the container (in MB). Must be at least 4 MB.

  - **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.

- **instance_type** [string] The EC2 instance type to deploy to. Only available for jobs running on kubernetes.
- **source** [string] The body/text of the script.
- **cancel_timeout** [integer] The amount of time (in seconds) to wait before forcibly terminating the script. When the script is cancelled, it is first sent a TERM signal. If the script is still running after the timeout, it is sent a KILL signal. Defaults to 0.

- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub.
- **partition_label** [string] The partition label used to run this object. Not generally available. Beware this attribute may be removed in the future.
- **running_as_id** [integer] The ID of the runner of this script.

put_r_git(*id*, ***, *git_ref='DEFAULT'*, *git_branch='DEFAULT'*, *git_path='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref_type='DEFAULT'*, *pull_from_git='DEFAULT'*)

Attach an item to a file in a git repo

> **Parameters**
>> **id** [integer] The ID of the file.
>> **git_ref** [string, optional] A git reference specifying an unambiguous version of the file. Can be a branch name, or the full or shortened SHA of a commit.
>> **git_branch** [string, optional] The git branch that the file is on.
>> **git_path** [string, optional] The path of the file in the repository.
>> **git_repo_url** [string, optional] The URL of the git repository.
>> **git_ref_type** [string, optional] Specifies if the file is versioned by branch or tag.
>> **pull_from_git** [boolean, optional] Automatically pull latest commit from git. Only works for scripts.
>
> **Returns**
>> *civis.response.Response*
>>> - **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
>>> - **git_branch** [string] The git branch that the file is on.
>>> - **git_path** [string] The path of the file in the repository.
>>> - **git_repo** [dict::]
>>>
>>>> - **id** [integer] The ID for this git repository.
>>>>
>>>> - **repo_url** [string] The URL for this git repository.
>>>>
>>>> - created_at : string/time
>>>>
>>>> - updated_at : string/time
>>>
>>> - **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
>>> - **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

put_r_projects(*id*, *project_id*)

Add an R Script to a project

> **Parameters**
>> **id** [integer] The ID of the R Script.
>> **project_id** [integer] The ID of the project.
>
> **Returns**
>> **None** Response code 204: success

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

Set the permissions groups has on this object

> **Parameters**
>> **id** [integer] The ID of the resource that is shared.
>> **group_ids** [list] An array of one or more group IDs.
>> **permission_level** [string] Options are: "read", "write", or "manage".
>> **share_email_body** [string, optional] Custom body text for e-mail sent on a share.

> > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
>
> > **Returns**
> > > *civis.response.Response*
> > > > • **readers** [dict::]
> > > > > – **users** [list::]
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **writers** [dict::]
> > > > > – **users** [list::]
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **owners** [dict::]
> > > > > – **users** [list::]
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object
>
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_ids** [list] An array of one or more user IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> > > *civis.response.Response*
> > > > • **readers** [dict::]
> > > > > – **users** [list::]
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string

> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **writers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **owners** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_r_transfer(*id*, *user_id*, *include_dependencies*, *, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

> Transfer ownership of this object to another user
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] ID of target user
> > >
> > > **include_dependencies** [boolean] Whether or not to give manage permissions on all dependencies
> > >
> > > **email_body** [string, optional] Custom body text for e-mail sent on transfer.
> > >
> > > **send_email** [boolean, optional] Send email to the target user of the transfer?
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > • **dependencies** [list::] Dependent objects for this object - object_type : string
> > >
> > > > Dependent object type
> > >
> > > > – **fco_type** [string] Human readable dependent object type
> > > >
> > > > – **id** [integer] Dependent object ID
> > > >
> > > > – **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > > >
> > > > – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user

–   **shared** [boolean] Whether dependent object was successfully
shared with target user

put_sql(*id*, *name*, *sql*, *remote_host_id*, *credential_id*, *, *parent_id='DEFAULT'*, *user_context='DEFAULT'*,
params='DEFAULT', *arguments='DEFAULT'*, *schedule='DEFAULT'*, *notifications='DEFAULT'*,
next_run_at='DEFAULT', *time_zone='DEFAULT'*, *target_project_id='DEFAULT'*,
csv_settings='DEFAULT', *running_as_id='DEFAULT'*)

Replace all attributes of this SQL script

> **Parameters**
>> **id** [integer] The ID for the script.
>> **name** [string] The name of the script.
>> **sql** [string] The raw SQL query for the script.
>> **remote_host_id** [integer] The remote host ID that this script will connect to.
>> **credential_id** [integer] The credential that this script will use.
>> **parent_id** [integer, optional] The ID of the parent job that will trigger this script
>> **user_context** [string, optional] "runner" or "author", who to execute the script as when
>>> run as a template.
>> **params** [list, optional::] A definition of the parameters this script accepts in the argu-
>>> ments field. - name : string
>>>> The variable's name as used within your code.
>>>
>>> • **label** [string] The label to present to users when asking them for the value.
>>> • **description** [string] A short sentence or fragment describing this param-
>>>>   eter to the end user.
>>> • **type** [string] The type of parameter.   Valid options: string,
>>>>   multi_line_string, integer, float, bool, file, table, database, cre-
>>>>   dential_aws, credential_redshift, or credential_custom
>>> • **required** [boolean] Whether this param is required.
>>> • **value** [string] The value you would like to set this param to. Setting this
>>>>   value makes this parameter a fixed param.
>>> • **default** [string] If an argument for this parameter is not defined, it will
>>>>   use this default value. Use true, True, t, y, yes, or 1 for true bool's
>>>>   or false, False, f, n, no, or 0 for false bool's. Cannot be used for
>>>>   parameters that are required or a credential type.
>>> • **allowed_values** [list] The possible values this parameter can take, effec-
>>>>   tively making this an enumerable parameter. Allowed values is an
>>>>   array of hashes of the following format: *{label: 'Import', 'value':
>>>>   'import'}*
>> **arguments** [dict, optional] Parameter-value pairs to use when running this script. Only
>>> settable if this script has defined parameters.
>> **schedule** [dict, optional::]
>>> • **scheduled** [boolean] If the item is scheduled.
>>> • **scheduled_days** [list] Days of the week, based on numeric value starting
>>>>   at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
>>> • **scheduled_hours** [list] Hours of the day it is scheduled on.
>>> • **scheduled_minutes** [list] Minutes of the day it is scheduled on.
>>> • **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes,
>>>>   number of times to run per hour.
>>> • **scheduled_days_of_month** [list] Days of the month it is scheduled on,
>>>>   mutually exclusive with scheduledDays.
>> **notifications** [dict, optional::]
>>> • **urls** [list] URLs to receive a POST request at job completion
>>> • **success_email_subject** [string] Custom subject line for success e-mail.
>>> • **success_email_body** [string] Custom body text for success e-mail, writ-
>>>>   ten in Markdown.

- **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.
- **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."
- **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.
- **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.
- **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.
- **success_on** [boolean] If success email notifications are on.
- **failure_on** [boolean] If failure email notifications are on.

next_run_at [string/time, optional] The time of the next scheduled run.

time_zone [string, optional] The time zone of this script.

target_project_id [integer, optional] Target project to which script outputs will be added.

csv_settings [dict, optional::]

- **include_header** [boolean] Whether or not to include headers in the output data. Default: true
- **compression** [string] The type of compression to use, if any, one of "none", "zip", or "gzip". Default: gzip
- **column_delimiter** [string] Which delimiter to use, one of "comma", "tab", or "pipe". Default: comma
- **unquoted** [boolean] Whether or not to quote fields. Default: false
- **force_multifile** [boolean] Whether or not the csv should be split into multiple files. Default: false
- **filename_prefix** [string] A user specified filename prefix for the output file to have. Default: null
- **max_file_size** [integer] The max file size, in MB, created files will be. Only available when force_multifile is true.

running_as_id [integer, optional] The ID of the runner of this script.

Returns

civis.response.Response

- **id** [integer] The ID for the script.
- **name** [string] The name of the script.
- **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
- **created_at** [string/time] The time this script was created.
- **updated_at** [string/time] The time the script was last updated.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

- **name** [string] The name of the project.
- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    - **details** [string] The details link to get more information about the script.

    - **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

- **scheduled_hours** [list] Hours of the day it is scheduled on.

- **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".

- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **sql** [string] The raw SQL query for the script.
- **expanded_arguments** [dict] Expanded arguments for use in injecting into different environments.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **code_preview** [string] The code that this script will run with arguments inserted.
- **csv_settings** [dict::]

    – **include_header** [boolean] Whether or not to include headers in the output data. Default: true

    – **compression** [string] The type of compression to use, if any, one of "none", "zip", or "gzip". Default: gzip

    – **column_delimiter** [string] Which delimiter to use, one of "comma", "tab", or "pipe". Default: comma

    – **unquoted** [boolean] Whether or not to quote fields. Default: false

    – **force_multifile** [boolean] Whether or not the csv should be split into multiple files. Default: false

    – **filename_prefix** [string] A user specified filename prefix for the output file to have. Default: null

    – **max_file_size** [integer] The max file size, in MB, created files will be. Only available when force_multifile is true.

- **running_as_id** [integer] The ID of the runner of this script.

put_sql_archive(*id*, *status*)

Update the archive status of this object

> **Parameters**
> > **id** [integer] The ID of the object.
> > **status** [boolean] The desired archived status of the object.
>
> **Returns**
> > *civis.response.Response*
> > - **id** [integer] The ID for the script.
> > - **name** [string] The name of the script.
> > - **type** [string] The type of the script (e.g SQL, Container, Python, R, JavaScript)
> > - **created_at** [string/time] The time this script was created.
> > - **updated_at** [string/time] The time the script was last updated.
> > - **author** [dict::]
> >
> >   – **id** [integer] The ID of this user.
> >
> >   – **name** [string] This user's name.
> >
> >   – **username** [string] This user's username.
> >
> >   – **initials** [string] This user's initials.
> >
> >   – **online** [boolean] Whether this user is online.

- **state** [string] The status of the script's last run.
- **finished_at** [string/time] The time that the script's last run finished.
- **category** [string] The category of the script.
- **projects** [list::] A list of projects containing the script. - id : integer

    The ID for the project.

    – **name** [string] The name of the project.

- **parent_id** [integer] The ID of the parent job that will trigger this script
- **user_context** [string] "runner" or "author", who to execute the script as when run as a template.
- **params** [list::] A definition of the parameters this script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    – **label** [string] The label to present to users when asking them for the value.

    – **description** [string] A short sentence or fragment describing this parameter to the end user.

    – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    – **required** [boolean] Whether this param is required.

    – **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    – **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    – **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **arguments** [dict] Parameter-value pairs to use when running this script. Only settable if this script has defined parameters.
- **is_template** [boolean] Whether others scripts use this one as a template.
- **published_as_template_id** [integer] The ID of the template that this script is backing.
- **from_template_id** [integer] The ID of the template this script uses, if any.
- **template_dependents_count** [integer] How many other scripts use this one as a template.
- **template_script_name** [string] The name of the template script.
- **links** [dict::]

    – **details** [string] The details link to get more information about the script.

    – **runs** [string] The runs link to get the run information list for this script.

- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **success_email_from_name** [string] Name from which success emails are sent; defaults to "Civis."

    - **success_email_reply_to** [string] Address for replies to success emails; defaults to the author of the job.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on.

    - **failure_on** [boolean] If failure email notifications are on.

- **running_as** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **next_run_at** [string/time] The time of the next scheduled run.
- **time_zone** [string] The time zone of this script.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

- **started_at** [string/time] The time that the run started.

- **finished_at** [string/time] The time that the run completed.

- **error** [string] The error message for this run, if present.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **hidden** [boolean] The hidden status of the item.
- **target_project_id** [integer] Target project to which script outputs will be added.
- **archived** [string] The archival status of the requested item(s).
- **sql** [string] The raw SQL query for the script.
- **expanded_arguments** [dict] Expanded arguments for use in injecting into different environments.
- **remote_host_id** [integer] The remote host ID that this script will connect to.
- **credential_id** [integer] The credential that this script will use.
- **code_preview** [string] The code that this script will run with arguments inserted.
- **csv_settings** [dict::]

    - **include_header** [boolean] Whether or not to include headers in the output data. Default: true

    - **compression** [string] The type of compression to use, if any, one of "none", "zip", or "gzip". Default: gzip

    - **column_delimiter** [string] Which delimiter to use, one of "comma", "tab", or "pipe". Default: comma

    - **unquoted** [boolean] Whether or not to quote fields. Default: false

    - **force_multifile** [boolean] Whether or not the csv should be split into multiple files. Default: false

    - **filename_prefix** [string] A user specified filename prefix for the output file to have. Default: null

    - **max_file_size** [integer] The max file size, in MB, created files will be. Only available when force_multifile is true.
- **running_as_id** [integer] The ID of the runner of this script.

put_sql_git(*id*, ***, *git_ref='DEFAULT'*, *git_branch='DEFAULT'*, *git_path='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref_type='DEFAULT'*, *pull_from_git='DEFAULT'*)

Attach an item to a file in a git repo

Parameters

id [integer] The ID of the file.

git_ref [string, optional] A git reference specifying an unambiguous version of the file. Can be a branch name, or the full or shortened SHA of a commit.

git_branch [string, optional] The git branch that the file is on.

git_path [string, optional] The path of the file in the repository.

git_repo_url [string, optional] The URL of the git repository.

git_ref_type [string, optional] Specifies if the file is versioned by branch or tag.

pull_from_git [boolean, optional] Automatically pull latest commit from git. Only works for scripts.

Returns

civis.response.Response

- **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
- **git_branch** [string] The git branch that the file is on.
- **git_path** [string] The path of the file in the repository.
- **git_repo** [dict::]

    - **id** [integer] The ID for this git repository.

    - **repo_url** [string] The URL for this git repository.

    - created_at : string/time

    - updated_at : string/time
- **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
- **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

put_sql_projects(*id*, *project_id*)

    Add a SQL script to a project

        Parameters

            id [integer] The ID of the SQL script.

            project_id [integer] The ID of the project.

        Returns

            None Response code 204: success

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

    Set the permissions groups has on this object

        Parameters

            id [integer] The ID of the resource that is shared.

            group_ids [list] An array of one or more group IDs.

            permission_level [string] Options are: "read", "write", or "manage".

            share_email_body [string, optional] Custom body text for e-mail sent on a share.

            send_shared_email [boolean, optional] Send email to the recipients of a share.

        Returns

        civis.response.Response

- **readers** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string
- **writers** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

$*$ name : string

- **owners** [dict::]

    - **users** [list::]

        $*$ id : integer

        $*$ name : string

    - **groups** [list::]

        $*$ id : integer

        $*$ name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

Set the permissions users have on this object

Parameters

id [integer] The ID of the resource that is shared.

user_ids [list] An array of one or more user IDs.

permission_level [string] Options are: "read", "write", or "manage".

share_email_body [string, optional] Custom body text for e-mail sent on a share.

send_shared_email [boolean, optional] Send email to the recipients of a share.

Returns

[*civis.response.Response*](#)

- **readers** [dict::]

    - **users** [list::]

        $*$ id : integer

        $*$ name : string

    - **groups** [list::]

        $*$ id : integer

        $*$ name : string

- **writers** [dict::]

    - **users** [list::]

        $*$ id : integer

        $*$ name : string

    - **groups** [list::]

        $*$ id : integer

        $*$ name : string

- **owners** [dict::]

    - **users** [list::]

        $*$ id : integer

        $*$ name : string

    - **groups** [list::]

* id : integer

* name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_sql_transfer(*id*, *user_id*, *include_dependencies*, ***, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

 Transfer ownership of this object to another user

  Parameters

   id [integer] The ID of the resource that is shared.

   user_id [integer] ID of target user

   include_dependencies [boolean] Whether or not to give manage permissions on all dependencies

   email_body [string, optional] Custom body text for e-mail sent on transfer.

   send_email [boolean, optional] Send email to the target user of the transfer?

  Returns

   civis.response.Response

- **dependencies** [list::] Dependent objects for this object - object_type : string

   Dependent object type

    – **fco_type** [string] Human readable dependent object type

    – **id** [integer] Dependent object ID

    – **name** [string] Dependent object name, or nil if the requesting user cannot read this object

    – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user

    – **shared** [boolean] Whether dependent object was successfully shared with target user

Search

class Search(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.search.list(...)
```

Methods

`list`(*[, query, type, offset, order, owner, ...])	Perform a search
`list_queries`(*[, search_string, ...])	Search queries that are not hidden
`list_types`()	List available search types

——

list(*, *query='DEFAULT'*, *type='DEFAULT'*, *offset='DEFAULT'*, *order='DEFAULT'*, *owner='DEFAULT'*, *limit='DEFAULT'*, *archived='DEFAULT'*, *last_run_state='DEFAULT'*)

Perform a search

> **Parameters**
>> **query** [string, optional] The search query.
>>
>> **type** [string, optional] The type for the search. It accepts a comma-separated list. Valid arguments are listed on the "GET /search/types" endpoint.
>>
>> **offset** [integer, optional] The offset for the search results.
>>
>> **order** [string, optional] The field on which to order the result set.
>>
>> **owner** [string, optional] The owner for the search.
>>
>> **limit** [integer, optional] Defaults to 10. Maximum allowed is 1000.
>>
>> **archived** [string, optional] If specified, return only results with the chosen archived status; either 'true', 'false', or 'all'. Defaults to 'false'.
>>
>> **last_run_state** [string, optional] The last run state of the job being searched for; either: 'queued', 'running', 'succeeded', 'failed', or 'cancelled'.
>
> **Returns**
>> `civis.response.Response`
>>> • **total_results** [integer] The number of items matching the search query.
>>> • **aggregations** [dict] Aggregations by owner and type for the search results.
>>> • **results** [list::] The items returned by the search. - score : number/float
>>>
>>>> The relevance score from the search request.
>>>>
>>>> – **type** [string] The type of the item.
>>>>
>>>> – **id** [integer] The ID of the item.
>>>>
>>>> – **name** [string] The name of the item.
>>>>
>>>> – **type_name** [string] The verbose name of the type.
>>>>
>>>> – **updated_at** [string/time] The time the item was last updated.
>>>>
>>>> – **owner** [string] The owner of the item.
>>>>
>>>> – **use_count** [integer] The use count of the item, if the item is a template.
>>>>
>>>> – **last_run_id** [integer] The last run id of the item, if the item is a job.
>>>>
>>>> – **last_run_state** [string] The last run state of the item, if the item is a job.
>>>>
>>>> – **last_run_start** [string/time] The last run start time of the item, if the item is a job.
>>>>
>>>> – **last_run_finish** [string/time] The last run finish time of the item, if the item is a job.

> > – **public** [boolean] The flag that indicates a template is available to all users.
> >
> > – **last_run_exception** [string] The exception of the item after the last run, if the item is a job.

list_queries(*, *search_string='DEFAULT'*, *database_id='DEFAULT'*, *credential_id='DEFAULT'*, *author_id='DEFAULT'*, *archived='DEFAULT'*, *state='DEFAULT'*, *started_before='DEFAULT'*, *started_after='DEFAULT'*, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)

> Search queries that are not hidden
>
> > **Parameters**
> >
> > > **search_string** [string, optional] Space delimited search terms for searching queries by their SQL. Supports wild card characters "?" for any single character, and "*" for zero or more characters.
> > >
> > > **database_id** [integer, optional] The database ID.
> > >
> > > **credential_id** [integer, optional] The credential ID.
> > >
> > > **author_id** [integer, optional] The author of the query.
> > >
> > > **archived** [boolean, optional] The archival status of the requested item(s). Defaults to false.
> > >
> > > **state** [array, optional] The state of the last run. One or more of queued, running, succeeded, failed, and cancelled.
> > >
> > > **started_before** [string, optional] An upper bound for the start date of the last run.
> > >
> > > **started_after** [string, optional] A lower bound for the start date of the last run.
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to 10. Maximum allowed is 1000.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to last_run_started_at. Must be one of: last_run_started_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**
> >
> > *civis.response.PaginatedResponse*
> >
> > > • **id** [integer] The query ID.
> > > • **database** [integer] The database ID.
> > > • **credential** [integer] The credential ID.
> > > • **sql** [string] The SQL executed by the query.
> > > • **author_id** [integer] The author of the query.
> > > • **archived** [boolean] The archival status of the requested item(s).
> > > • created_at : string/time
> > > • updated_at : string/time
> > > • **last_run** [dict::]
> > >
> > > > – id : integer
> > > >
> > > > – **state** [string] The state of the run. One of queued, running, succeeded, failed, and cancelled.
> > > >
> > > > – **started_at** [string/time] The time that the run started.
> > > >
> > > > – **finished_at** [string/time] The time that the run completed.
> > > >
> > > > – **error** [string] The error message for this run, if present.

list_types()

> List available search types

> > **Returns**

> > > *civis.response.Response*

> > > > • **type** [string] The name of the item type.

Services

class Services(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.services.list(...)
```

Methods

delete_deployments(service_id, deployment_id)	Delete a Service deployment
delete_projects(id, project_id)	Remove a Service 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_tokens(id, token_id)	Revoke a token by id
get(id)	Get a Service
get_deployments(service_id, deployment_id)	Get details about a Service deployment
list(*[, hidden, archived, author, status, ...])	List Services
list_dependencies(id, *[, user_id])	List dependent objects for this object
list_deployments(service_id, *[, ...])	List deployments for a Service
list_deployments_logs(id, deployment_id, *)	Get the logs for a Service deployment
list_projects(id, *[, hidden])	List the projects a Service belongs to
list_shares(id)	List users and groups permissioned on this object
list_tokens(id)	List tokens
patch(id, *[, name, description, ...])	Update some attributes of this Service
post(*[, name, description, type, ...])	Create a Service
post_clone(id)	Clone this Service
post_deployments(service_id, *[, deployment_id])	Deploy a Service
post_redeploy(service_id, *[, deployment_id])	Redeploy a Service
post_tokens(id, name, *[, machine_token, ...])	Create a new long-lived service token
put(id, *[, name, description, ...])	Replace all attributes of this Service
put_archive(id, status)	Update the archive status of this object
put_projects(id, project_id)	Add a Service to a project
put_shares_groups(id, group_ids, ...[, ...])	Set the permissions groups has on this object
put_shares_users(id, user_ids, ...[, ...])	Set the permissions users have on this object
put_transfer(id, user_id, ...[, email_body, ...])	Transfer ownership of this object to another user

—

delete_deployments(*service_id*, *deployment_id*)

> Delete a Service deployment
>
> > **Parameters**
> >
> > > **service_id** [integer] The ID of the owning Service
> > >
> > > **deployment_id** [integer] The ID for this deployment
> >
> > **Returns**
> >
> > > **None** Response code 204: success

delete_projects(*id*, *project_id*)

> Remove a Service from a project
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the Service.
> > >
> > > **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] The ID of the resource that is shared.
> > >
> > > **group_id** [integer] The 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] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] The ID of the user.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

delete_tokens(*id*, *token_id*)

> Revoke a token by id
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the service.
> > >
> > > **token_id** [integer] The ID of the token.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

get(*id*)

> Get a Service
>
> > **Parameters**
> >
> > > **id** [integer]
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > - **id** [integer] The ID for this Service.
> > > > - **name** [string] The name of this Service.
> > > > - **description** [string] The description of this Service.
> > > > - **user** [dict::]
> > > >
> > > > > - **id** [integer] The ID of this user.
> > > > >
> > > > > - **name** [string] This user's name.
> > > > >
> > > > > - **username** [string] This user's username.

> > > > – **initials** [string] This user's initials.
> > > >
> > > > – **online** [boolean] Whether this user is online.

- **type** [string] The type of this Service
- **docker_image_name** [string] The name of the docker image to pull from DockerHub.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
- **schedule** [dict::]

> > > > – **runtime_plan** [string] Only affects the service when deployed. On Demand means that the service will be turned on when viewed and automatically turned off after periods of inactivity. Specific Times means the service will be on when scheduled. Always On means the deployed service will always be on.
> > > >
> > > > – **recurrences** [list::] List of day-hour combinations this item is scheduled for - scheduled_days : list
> > > >
> > > > > > Days it is scheduled on, based on numeric value starting at 0 for Sunday
> > > > >
> > > > > > ∗ **scheduled_hours** [list] Hours it is scheduled on

- time_zone : string
- **replicas** [integer] The number of Service replicas to deploy. When maxReplicas is set, this field defines the minimum number of replicas to deploy.
- **max_replicas** [integer] The maximum number of Service replicas to deploy. Defining this field enables autoscaling.
- **instance_type** [string] The EC2 instance type to deploy to.
- **memory** [integer] The amount of memory allocated to each replica of the Service.
- **cpu** [integer] The amount of cpu allocated to each replica of the the Service.
- created_at : string/time
- updated_at : string/time
- **credentials** [list] A list of credential IDs to pass to the Service.
- **permission_set_id** [integer] The ID of the associated permission set, if any.
- **git_repo_url** [string] The url for the git repo where the Service code lives.
- **git_repo_ref** [string] The git reference to use when pulling code from the repo.
- **git_path_dir** [string] The path to the Service code within the git repo. If unspecified, the root directory will be used.
- **report_id** [integer] The ID of the associated report.
- **current_deployment** [dict::]

> > > – **deployment_id** [integer] The ID for this deployment.
> > >
> > > – **user_id** [integer] The ID of the owner.
> > >
> > > – **host** [string] Domain of the deployment.
> > >
> > > – **name** [string] Name of the deployment.
> > >
> > > – **docker_image_name** [string] The name of the docker image to pull from DockerHub.

- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).

- **display_url** [string] A signed URL for viewing the deployed item.

- **instance_type** [string] The EC2 instance type requested for the deployment.

- **memory** [integer] The memory allocated to the deployment, in MB.

- **cpu** [integer] The cpu allocated to the deployment, in millicores.

- **state** [string] The state of the deployment.

- **state_message** [string] A detailed description of the state.

- **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.

- **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.

- created_at : string/time

- updated_at : string/time

- **service_id** [integer] The ID of owning Service

- **current_url** [string] The URL that the service is hosted at.
- **environment_variables** [dict] Environment Variables to be passed into the Service.
- **notifications** [dict::]

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the service fails.

    - **failure_on** [boolean] If failure email notifications are on

- **partition_label** [string] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.

get_deployments(*service_id*, *deployment_id*)

Get details about a Service deployment

    Parameters

        service_id [integer] The ID of the owning Service

        deployment_id [integer] The ID for this deployment

    Returns

        civis.response.Response

- **deployment_id** [integer] The ID for this deployment.
- **user_id** [integer] The ID of the owner.
- **host** [string] Domain of the deployment.
- **name** [string] Name of the deployment.

> - **docker_image_name** [string] The name of the docker image to pull from DockerHub.
> - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
> - **display_url** [string] A signed URL for viewing the deployed item.
> - **instance_type** [string] The EC2 instance type requested for the deployment.
> - **memory** [integer] The memory allocated to the deployment, in MB.
> - **cpu** [integer] The cpu allocated to the deployment, in millicores.
> - **state** [string] The state of the deployment.
> - **state_message** [string] A detailed description of the state.
> - **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.
> - **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.
> - created_at : string/time
> - updated_at : string/time
> - **service_id** [integer] The ID of owning Service

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

List Services

Parameters

hidden [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.

archived [string, optional] The archival status of the requested item(s).

author [string, optional] If specified, return items from any of these authors. It accepts a comma- separated list of user IDs.

status [string, optional] If specified, returns Services with one of these statuses. It accepts a comma-separated list, possible values are 'running', 'idle'.

limit [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 50.

page_num [integer, optional] Page number of the results to return. Defaults to the first page, 1.

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

order_dir [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

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

Returns

civis.response.PaginatedResponse

- **id** [integer] The ID for this Service.
- **name** [string] The name of this Service.
- **description** [string] The description of this Service.
- **user** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **type** [string] The type of this Service
- created_at : string/time
- updated_at : string/time
- **git_repo_url** [string] The url for the git repo where the Service code lives.
- **git_repo_ref** [string] The git reference to use when pulling code from the repo.
- **git_path_dir** [string] The path to the Service code within the git repo. If unspecified, the root directory will be used.
- **current_deployment** [dict::]

    - **deployment_id** [integer] The ID for this deployment.

    - **user_id** [integer] The ID of the owner.

    - **host** [string] Domain of the deployment.

    - **name** [string] Name of the deployment.

    - **docker_image_name** [string] The name of the docker image to pull from DockerHub.

    - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).

    - **instance_type** [string] The EC2 instance type requested for the deployment.

    - **memory** [integer] The memory allocated to the deployment, in MB.

    - **cpu** [integer] The cpu allocated to the deployment, in millicores.

    - **state** [string] The state of the deployment.

    - **state_message** [string] A detailed description of the state.

    - **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.

    - **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.

    - created_at : string/time

    - updated_at : string/time

    - **service_id** [integer] The ID of owning Service
- **archived** [string] The archival status of the requested item(s).

list_dependencies(*id*, ***, *user_id='DEFAULT'*)

    List dependent objects for this object

        Parameters

            id [integer] The ID of the resource that is shared.

            user_id [integer, optional] ID of target user

        Returns

            civis.response.Response

- **object_type** [string] Dependent object type
- **fco_type** [string] Human readable dependent object type

- **id** [integer] Dependent object ID
- **name** [string] Dependent object name, or nil if the requesting user cannot read this object
- **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
- **shareable** [boolean] Whether or not the requesting user can share this object.

list_deployments(*service_id*, *, *deployment_id='DEFAULT'*, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)

> List deployments for a Service
>
> > **Parameters**
> >
> > > **service_id** [integer] The ID of the owning Service
> > >
> > > **deployment_id** [integer, optional] The ID for this deployment
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 50.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to created_at. Must be one of: created_at.
> > >
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > >
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> >
> > **Returns**
> >
> > *civis.response.PaginatedResponse*
> >
> > > - **deployment_id** [integer] The ID for this deployment.
> > > - **user_id** [integer] The ID of the owner.
> > > - **host** [string] Domain of the deployment.
> > > - **name** [string] Name of the deployment.
> > > - **docker_image_name** [string] The name of the docker image to pull from DockerHub.
> > > - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
> > > - **instance_type** [string] The EC2 instance type requested for the deployment.
> > > - **memory** [integer] The memory allocated to the deployment, in MB.
> > > - **cpu** [integer] The cpu allocated to the deployment, in millicores.
> > > - **state** [string] The state of the deployment.
> > > - **state_message** [string] A detailed description of the state.
> > > - **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.
> > > - **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.
> > > - created_at : string/time
> > > - updated_at : string/time
> > > - **service_id** [integer] The ID of owning Service

list_deployments_logs(*id*, *deployment_id*, *, *start_at='DEFAULT'*, *end_at='DEFAULT'*, *limit='DEFAULT'*)

> Get the logs for a Service deployment
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the owning Service.
> > >
> > > **deployment_id** [integer] The ID for this deployment.

> > **start_at** [string, optional] Log entries with a lower timestamp will be omitted.
> >
> > **end_at** [string, optional] Log entries with a higher timestamp will be omitted.
> >
> > **limit** [integer, optional] The maximum number of log messages to return. Default of 10000.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **message** [string] The log message.
> > > > • **stream** [string] The stream of the log. One of "stdout", "stderr".
> > > > • **created_at** [string/date-time] The time the log was created.
> > > > • **source** [string] The source of the log. One of "system", "user".

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

> List the projects a Service belongs to
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the Service.
> > >
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **id** [integer] The ID for this project.
> > > > • **author** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > • **name** [string] The name of this project.
> > > > • **description** [string] A description of the project.
> > > > • **users** [list::] Users who can see the project. - id : integer
> > > >
> > > > > The ID of this user.
> > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > • auto_share : boolean
> > > > • created_at : string/time
> > > > • updated_at : string/time
> > > > • **archived** [string] The archival status of the requested item(s).

list_shares(*id*)

> List users and groups permissioned on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **readers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer

> > > > > > > > > > ∗ name : string

> > > > > > > > – **groups** [list::]

> > > > > > > > > > ∗ id : integer

> > > > > > > > > > ∗ name : string

> > > > > > • **writers** [dict::]

> > > > > > > > – **users** [list::]

> > > > > > > > > > ∗ id : integer

> > > > > > > > > > ∗ name : string

> > > > > > > > – **groups** [list::]

> > > > > > > > > > ∗ id : integer

> > > > > > > > > > ∗ name : string

> > > > > > • **owners** [dict::]

> > > > > > > > – **users** [list::]

> > > > > > > > > > ∗ id : integer

> > > > > > > > > > ∗ name : string

> > > > > > > > – **groups** [list::]

> > > > > > > > > > ∗ id : integer

> > > > > > > > > > ∗ name : string

> > > > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_tokens(*id*)

> List tokens

> > **Parameters**

> > > **id** [integer] The ID of the service.

> > **Returns**

> > [*civis.response.Response*](#)

> > > • **id** [integer] The ID of the token.
> > > • **name** [string] The name of the token.
> > > • **user** [dict::]

> > > > – **id** [integer] The ID of this user.

> > > > – **name** [string] This user's name.

> > > > – **username** [string] This user's username.

> > > > – **initials** [string] This user's initials.

> > > > – **online** [boolean] Whether this user is online.

> > > • **machine_token** [boolean] If true, this token is not tied to a particular user.
> > > • **expires_at** [string/date-time] The date and time when the token expires.
> > > • **created_at** [string/time] The date and time when the token was created.

patch(*id*, ***, *name='DEFAULT'*, *description='DEFAULT'*, *docker_image_name='DEFAULT'*,
      docker_image_tag='DEFAULT', *schedule='DEFAULT'*, *replicas='DEFAULT'*,
      max_replicas='DEFAULT', *instance_type='DEFAULT'*, *memory='DEFAULT'*, *cpu='DEFAULT'*,
      credentials='DEFAULT', *permission_set_id='DEFAULT'*, *git_repo_url='DEFAULT'*,
      git_repo_ref='DEFAULT', *git_path_dir='DEFAULT'*, *environment_variables='DEFAULT'*,
      notifications='DEFAULT', *partition_label='DEFAULT'*)

Update some attributes of this Service

Parameters

id [integer] The ID for this Service.

name [string, optional] The name of this Service.

description [string, optional] The description of this Service.

docker_image_name [string, optional] The name of the docker image to pull from DockerHub.

docker_image_tag [string, optional] The tag of the docker image to pull from DockerHub (default: latest).

schedule [dict, optional::]

- **runtime_plan** [string] Only affects the service when deployed. On Demand means that the service will be turned on when viewed and automatically turned off after periods of inactivity. Specific Times means the service will be on when scheduled. Always On means the deployed service will always be on.

- **recurrences** [list::] List of day-hour combinations this item is scheduled for - scheduled_days : list

      Days it is scheduled on, based on numeric value starting at 0 for Sunday

      – **scheduled_hours** [list] Hours it is scheduled on

replicas [integer, optional] The number of Service replicas to deploy. When maxReplicas is set, this field defines the minimum number of replicas to deploy.

max_replicas [integer, optional] The maximum number of Service replicas to deploy. Defining this field enables autoscaling.

instance_type [string, optional] The EC2 instance type to deploy to.

memory [integer, optional] The amount of memory allocated to each replica of the Service.

cpu [integer, optional] The amount of cpu allocated to each replica of the the Service.

credentials [list, optional] A list of credential IDs to pass to the Service.

permission_set_id [integer, optional] The ID of the associated permission set, if any.

git_repo_url [string, optional] The url for the git repo where the Service code lives.

git_repo_ref [string, optional] The git reference to use when pulling code from the repo.

git_path_dir [string, optional] The path to the Service code within the git repo. If unspecified, the root directory will be used.

environment_variables [dict, optional] Environment Variables to be passed into the Service.

notifications [dict, optional::]

- **failure_email_addresses** [list] Addresses to notify by e-mail when the service fails.

- **failure_on** [boolean] If failure email notifications are on

partition_label [string, optional] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.

Returns

civis.response.Response

- **id** [integer] The ID for this Service.
- **name** [string] The name of this Service.
- **description** [string] The description of this Service.
- **user** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- **type** [string] The type of this Service
- **docker_image_name** [string] The name of the docker image to pull from DockerHub.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
- **schedule** [dict::]

    – **runtime_plan** [string] Only affects the service when deployed. On Demand means that the service will be turned on when viewed and automatically turned off after periods of inactivity. Specific Times means the service will be on when scheduled. Always On means the deployed service will always be on.

    – **recurrences** [list::] List of day-hour combinations this item is scheduled for - scheduled_days : list

        Days it is scheduled on, based on numeric value starting at 0 for Sunday

        * **scheduled_hours** [list] Hours it is scheduled on
- time_zone : string
- **replicas** [integer] The number of Service replicas to deploy. When maxReplicas is set, this field defines the minimum number of replicas to deploy.
- **max_replicas** [integer] The maximum number of Service replicas to deploy. Defining this field enables autoscaling.
- **instance_type** [string] The EC2 instance type to deploy to.
- **memory** [integer] The amount of memory allocated to each replica of the Service.
- **cpu** [integer] The amount of cpu allocated to each replica of the the Service.
- created_at : string/time
- updated_at : string/time
- **credentials** [list] A list of credential IDs to pass to the Service.
- **permission_set_id** [integer] The ID of the associated permission set, if any.
- **git_repo_url** [string] The url for the git repo where the Service code lives.
- **git_repo_ref** [string] The git reference to use when pulling code from the repo.
- **git_path_dir** [string] The path to the Service code within the git repo. If unspecified, the root directory will be used.
- **report_id** [integer] The ID of the associated report.

- **current_deployment** [dict::]

    - **deployment_id** [integer] The ID for this deployment.

    - **user_id** [integer] The ID of the owner.

    - **host** [string] Domain of the deployment.

    - **name** [string] Name of the deployment.

    - **docker_image_name** [string] The name of the docker image to pull from DockerHub.

    - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).

    - **display_url** [string] A signed URL for viewing the deployed item.

    - **instance_type** [string] The EC2 instance type requested for the deployment.

    - **memory** [integer] The memory allocated to the deployment, in MB.

    - **cpu** [integer] The cpu allocated to the deployment, in millicores.

    - **state** [string] The state of the deployment.

    - **state_message** [string] A detailed description of the state.

    - **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.

    - **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.

    - created_at : string/time

    - updated_at : string/time

    - **service_id** [integer] The ID of owning Service
- **current_url** [string] The URL that the service is hosted at.
- **environment_variables** [dict] Environment Variables to be passed into the Service.
- **notifications** [dict::]

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the service fails.

    - **failure_on** [boolean] If failure email notifications are on
- **partition_label** [string] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.

post(*, *name='DEFAULT'*, *description='DEFAULT'*, *type='DEFAULT'*, *docker_image_name='DEFAULT'*, *docker_image_tag='DEFAULT'*, *schedule='DEFAULT'*, *replicas='DEFAULT'*, *max_replicas='DEFAULT'*, *instance_type='DEFAULT'*, *memory='DEFAULT'*, *cpu='DEFAULT'*, *credentials='DEFAULT'*, *permission_set_id='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_repo_ref='DEFAULT'*, *git_path_dir='DEFAULT'*, *environment_variables='DEFAULT'*, *notifications='DEFAULT'*, *partition_label='DEFAULT'*, *hidden='DEFAULT'*)

Create a Service

Parameters

name [string, optional] The name of this Service.

description [string, optional] The description of this Service.

type [string, optional] The type of this Service

docker_image_name [string, optional] The name of the docker image to pull from DockerHub.

docker_image_tag [string, optional] The tag of the docker image to pull from DockerHub (default: latest).

schedule [dict, optional::]

- **runtime_plan** [string] Only affects the service when deployed. On Demand means that the service will be turned on when viewed and automatically turned off after periods of inactivity. Specific Times means the service will be on when scheduled. Always On means the deployed service will always be on.

- **recurrences** [list::] List of day-hour combinations this item is scheduled for - scheduled_days : list

   Days it is scheduled on, based on numeric value starting at 0 for Sunday

   – **scheduled_hours** [list] Hours it is scheduled on

replicas [integer, optional] The number of Service replicas to deploy. When maxReplicas is set, this field defines the minimum number of replicas to deploy.

max_replicas [integer, optional] The maximum number of Service replicas to deploy. Defining this field enables autoscaling.

instance_type [string, optional] The EC2 instance type to deploy to.

memory [integer, optional] The amount of memory allocated to each replica of the Service.

cpu [integer, optional] The amount of cpu allocated to each replica of the the Service.

credentials [list, optional] A list of credential IDs to pass to the Service.

permission_set_id [integer, optional] The ID of the associated permission set, if any.

git_repo_url [string, optional] The url for the git repo where the Service code lives.

git_repo_ref [string, optional] The git reference to use when pulling code from the repo.

git_path_dir [string, optional] The path to the Service code within the git repo. If unspecified, the root directory will be used.

environment_variables [dict, optional] Environment Variables to be passed into the Service.

notifications [dict, optional::]

- **failure_email_addresses** [list] Addresses to notify by e-mail when the service fails.

- **failure_on** [boolean] If failure email notifications are on

partition_label [string, optional] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.

hidden [boolean, optional] The hidden status of the item.

Returns

`civis.response.Response`

- **id** [integer] The ID for this Service.
- **name** [string] The name of this Service.
- **description** [string] The description of this Service.
- **user** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **type** [string] The type of this Service
- **docker_image_name** [string] The name of the docker image to pull from DockerHub.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
- **schedule** [dict::]

  - **runtime_plan** [string] Only affects the service when deployed. On Demand means that the service will be turned on when viewed and automatically turned off after periods of inactivity. Specific Times means the service will be on when scheduled. Always On means the deployed service will always be on.

  - **recurrences** [list::] List of day-hour combinations this item is scheduled for - scheduled_days : list

    Days it is scheduled on, based on numeric value starting at 0 for Sunday

    * **scheduled_hours** [list] Hours it is scheduled on
- time_zone : string
- **replicas** [integer] The number of Service replicas to deploy. When maxReplicas is set, this field defines the minimum number of replicas to deploy.
- **max_replicas** [integer] The maximum number of Service replicas to deploy. Defining this field enables autoscaling.
- **instance_type** [string] The EC2 instance type to deploy to.
- **memory** [integer] The amount of memory allocated to each replica of the Service.
- **cpu** [integer] The amount of cpu allocated to each replica of the the Service.
- created_at : string/time
- updated_at : string/time
- **credentials** [list] A list of credential IDs to pass to the Service.
- **permission_set_id** [integer] The ID of the associated permission set, if any.
- **git_repo_url** [string] The url for the git repo where the Service code lives.
- **git_repo_ref** [string] The git reference to use when pulling code from the repo.
- **git_path_dir** [string] The path to the Service code within the git repo. If unspecified, the root directory will be used.

- **report_id** [integer] The ID of the associated report.
- **current_deployment** [dict::]

    - **deployment_id** [integer] The ID for this deployment.

    - **user_id** [integer] The ID of the owner.

    - **host** [string] Domain of the deployment.

    - **name** [string] Name of the deployment.

    - **docker_image_name** [string] The name of the docker image to pull from DockerHub.

    - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).

    - **display_url** [string] A signed URL for viewing the deployed item.

    - **instance_type** [string] The EC2 instance type requested for the deployment.

    - **memory** [integer] The memory allocated to the deployment, in MB.

    - **cpu** [integer] The cpu allocated to the deployment, in millicores.

    - **state** [string] The state of the deployment.

    - **state_message** [string] A detailed description of the state.

    - **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.

    - **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.

    - created_at : string/time

    - updated_at : string/time

    - **service_id** [integer] The ID of owning Service
- **current_url** [string] The URL that the service is hosted at.
- **environment_variables** [dict] Environment Variables to be passed into the Service.
- **notifications** [dict::]

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the service fails.

    - **failure_on** [boolean] If failure email notifications are on
- **partition_label** [string] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.

post_clone(*id*)

Clone this Service

> **Parameters**
>> **id** [integer]

> **Returns**

>> [*civis.response.Response*](civis.response.Response)

>>> - **id** [integer] The ID for this Service.
>>> - **name** [string] The name of this Service.
>>> - **description** [string] The description of this Service.
>>> - **user** [dict::]
>>>
>>>> – **id** [integer] The ID of this user.
>>>>
>>>> – **name** [string] This user's name.
>>>>
>>>> – **username** [string] This user's username.
>>>>
>>>> – **initials** [string] This user's initials.
>>>>
>>>> – **online** [boolean] Whether this user is online.
>>>
>>> - **type** [string] The type of this Service
>>> - **docker_image_name** [string] The name of the docker image to pull from DockerHub.
>>> - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
>>> - **schedule** [dict::]
>>>
>>>> – **runtime_plan** [string] Only affects the service when deployed. On Demand means that the service will be turned on when viewed and automatically turned off after periods of inactivity. Specific Times means the service will be on when scheduled. Always On means the deployed service will always be on.
>>>>
>>>> – **recurrences** [list::] List of day-hour combinations this item is scheduled for - scheduled_days : list
>>>>
>>>>> Days it is scheduled on, based on numeric value starting at 0 for Sunday
>>>>>
>>>>> ∗ **scheduled_hours** [list] Hours it is scheduled on
>>>
>>> - time_zone : string
>>> - **replicas** [integer] The number of Service replicas to deploy. When maxReplicas is set, this field defines the minimum number of replicas to deploy.
>>> - **max_replicas** [integer] The maximum number of Service replicas to deploy. Defining this field enables autoscaling.
>>> - **instance_type** [string] The EC2 instance type to deploy to.
>>> - **memory** [integer] The amount of memory allocated to each replica of the Service.
>>> - **cpu** [integer] The amount of cpu allocated to each replica of the the Service.
>>> - created_at : string/time
>>> - updated_at : string/time
>>> - **credentials** [list] A list of credential IDs to pass to the Service.
>>> - **permission_set_id** [integer] The ID of the associated permission set, if any.

5.5. API Client 657

- **git_repo_url** [string] The url for the git repo where the Service code lives.
- **git_repo_ref** [string] The git reference to use when pulling code from the repo.
- **git_path_dir** [string] The path to the Service code within the git repo. If unspecified, the root directory will be used.
- **report_id** [integer] The ID of the associated report.
- **current_deployment** [dict::]

    – **deployment_id** [integer] The ID for this deployment.

    – **user_id** [integer] The ID of the owner.

    – **host** [string] Domain of the deployment.

    – **name** [string] Name of the deployment.

    – **docker_image_name** [string] The name of the docker image to pull from DockerHub.

    – **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).

    – **display_url** [string] A signed URL for viewing the deployed item.

    – **instance_type** [string] The EC2 instance type requested for the deployment.

    – **memory** [integer] The memory allocated to the deployment, in MB.

    – **cpu** [integer] The cpu allocated to the deployment, in millicores.

    – **state** [string] The state of the deployment.

    – **state_message** [string] A detailed description of the state.

    – **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.

    – **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.

    – created_at : string/time

    – updated_at : string/time

    – **service_id** [integer] The ID of owning Service
- **current_url** [string] The URL that the service is hosted at.
- **environment_variables** [dict] Environment Variables to be passed into the Service.
- **notifications** [dict::]

    – **failure_email_addresses** [list] Addresses to notify by e-mail when the service fails.

    – **failure_on** [boolean] If failure email notifications are on
- **partition_label** [string] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.

- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.

post_deployments(*service_id*, *, *deployment_id='DEFAULT'*)

    Deploy a Service

        Parameters

            service_id [integer] The ID of the owning Service

            deployment_id [integer, optional] The ID for this deployment

        Returns

            civis.response.Response

- **deployment_id** [integer] The ID for this deployment.
- **user_id** [integer] The ID of the owner.
- **host** [string] Domain of the deployment.
- **name** [string] Name of the deployment.
- **docker_image_name** [string] The name of the docker image to pull from DockerHub.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
- **display_url** [string] A signed URL for viewing the deployed item.
- **instance_type** [string] The EC2 instance type requested for the deployment.
- **memory** [integer] The memory allocated to the deployment, in MB.
- **cpu** [integer] The cpu allocated to the deployment, in millicores.
- **state** [string] The state of the deployment.
- **state_message** [string] A detailed description of the state.
- **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.
- **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.
- created_at : string/time
- updated_at : string/time
- **service_id** [integer] The ID of owning Service

post_redeploy(*service_id*, *, *deployment_id='DEFAULT'*)

    Redeploy a Service

        Parameters

            service_id [integer] The ID of the owning Service

            deployment_id [integer, optional] The ID for this deployment

        Returns

            civis.response.Response

- **deployment_id** [integer] The ID for this deployment.
- **user_id** [integer] The ID of the owner.
- **host** [string] Domain of the deployment.
- **name** [string] Name of the deployment.
- **docker_image_name** [string] The name of the docker image to pull from DockerHub.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
- **display_url** [string] A signed URL for viewing the deployed item.
- **instance_type** [string] The EC2 instance type requested for the deployment.
- **memory** [integer] The memory allocated to the deployment, in MB.
- **cpu** [integer] The cpu allocated to the deployment, in millicores.

> - **state** [string] The state of the deployment.
> - **state_message** [string] A detailed description of the state.
> - **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.
> - **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.
> - created_at : string/time
> - updated_at : string/time
> - **service_id** [integer] The ID of owning Service

post_tokens(*id*, *name*, *, *machine_token='DEFAULT'*, *expires_in='DEFAULT'*)

Create a new long-lived service token

> **Parameters**
>> **id** [integer] The ID of the service.
>>
>> **name** [string] The name of the token.
>>
>> **machine_token** [boolean, optional] If true, create a compact token with no user information.
>>
>> **expires_in** [integer, optional] The number of seconds until the token should expire
>
> **Returns**
>> *civis.response.Response*
>>
>>> - **id** [integer] The ID of the token.
>>> - **name** [string] The name of the token.
>>> - **user** [dict::]
>>>
>>>> - **id** [integer] The ID of this user.
>>>>
>>>> - **name** [string] This user's name.
>>>>
>>>> - **username** [string] This user's username.
>>>>
>>>> - **initials** [string] This user's initials.
>>>>
>>>> - **online** [boolean] Whether this user is online.
>>>
>>> - **machine_token** [boolean] If true, this token is not tied to a particular user.
>>> - **expires_at** [string/date-time] The date and time when the token expires.
>>> - **created_at** [string/time] The date and time when the token was created.
>>> - **token** [string] The value of the token. Only returned when the token is first created.

put(*id*, *, *name='DEFAULT'*, *description='DEFAULT'*, *docker_image_name='DEFAULT'*, *docker_image_tag='DEFAULT'*, *schedule='DEFAULT'*, *replicas='DEFAULT'*, *max_replicas='DEFAULT'*, *instance_type='DEFAULT'*, *memory='DEFAULT'*, *cpu='DEFAULT'*, *credentials='DEFAULT'*, *permission_set_id='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_repo_ref='DEFAULT'*, *git_path_dir='DEFAULT'*, *environment_variables='DEFAULT'*, *notifications='DEFAULT'*, *partition_label='DEFAULT'*)

Replace all attributes of this Service

> **Parameters**
>> **id** [integer] The ID for this Service.
>>
>> **name** [string, optional] The name of this Service.
>>
>> **description** [string, optional] The description of this Service.
>>
>> **docker_image_name** [string, optional] The name of the docker image to pull from DockerHub.
>>
>> **docker_image_tag** [string, optional] The tag of the docker image to pull from DockerHub (default: latest).
>>
>> **schedule** [dict, optional::]
>>
>>> - **runtime_plan** [string] Only affects the service when deployed. On Demand means that the service will be turned on when viewed and

> > automatically turned off after periods of inactivity. Specific Times means the service will be on when scheduled. Always On means the deployed service will always be on.

- **recurrences** [list::] List of day-hour combinations this item is scheduled for - scheduled_days : list

> > > Days it is scheduled on, based on numeric value starting at 0 for Sunday

> > – **scheduled_hours** [list] Hours it is scheduled on

replicas [integer, optional] The number of Service replicas to deploy. When maxReplicas is set, this field defines the minimum number of replicas to deploy.

max_replicas [integer, optional] The maximum number of Service replicas to deploy. Defining this field enables autoscaling.

instance_type [string, optional] The EC2 instance type to deploy to.

memory [integer, optional] The amount of memory allocated to each replica of the Service.

cpu [integer, optional] The amount of cpu allocated to each replica of the the Service.

credentials [list, optional] A list of credential IDs to pass to the Service.

permission_set_id [integer, optional] The ID of the associated permission set, if any.

git_repo_url [string, optional] The url for the git repo where the Service code lives.

git_repo_ref [string, optional] The git reference to use when pulling code from the repo.

git_path_dir [string, optional] The path to the Service code within the git repo. If unspecified, the root directory will be used.

environment_variables [dict, optional] Environment Variables to be passed into the Service.

notifications [dict, optional::]

> - **failure_email_addresses** [list] Addresses to notify by e-mail when the service fails.
> - **failure_on** [boolean] If failure email notifications are on

partition_label [string, optional] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.

Returns

civis.response.Response

> - **id** [integer] The ID for this Service.
> - **name** [string] The name of this Service.
> - **description** [string] The description of this Service.
> - **user** [dict::]
>
> > – **id** [integer] The ID of this user.
> >
> > – **name** [string] This user's name.
> >
> > – **username** [string] This user's username.
> >
> > – **initials** [string] This user's initials.
> >
> > – **online** [boolean] Whether this user is online.
>
> - **type** [string] The type of this Service
> - **docker_image_name** [string] The name of the docker image to pull from DockerHub.
> - **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
> - **schedule** [dict::]

> > > – **runtime_plan** [string] Only affects the service when deployed. On Demand means that the service will be turned on when viewed and automatically turned off after periods of inactivity. Specific Times means the service will be on when scheduled. Always On means the deployed service will always be on.
> > >
> > > – **recurrences** [list::] List of day-hour combinations this item is scheduled for - scheduled_days : list
> > >
> > > > Days it is scheduled on, based on numeric value starting at 0 for Sunday
> > >
> > > > * **scheduled_hours** [list] Hours it is scheduled on

- time_zone : string
- **replicas** [integer] The number of Service replicas to deploy. When maxReplicas is set, this field defines the minimum number of replicas to deploy.
- **max_replicas** [integer] The maximum number of Service replicas to deploy. Defining this field enables autoscaling.
- **instance_type** [string] The EC2 instance type to deploy to.
- **memory** [integer] The amount of memory allocated to each replica of the Service.
- **cpu** [integer] The amount of cpu allocated to each replica of the the Service.
- created_at : string/time
- updated_at : string/time
- **credentials** [list] A list of credential IDs to pass to the Service.
- **permission_set_id** [integer] The ID of the associated permission set, if any.
- **git_repo_url** [string] The url for the git repo where the Service code lives.
- **git_repo_ref** [string] The git reference to use when pulling code from the repo.
- **git_path_dir** [string] The path to the Service code within the git repo. If unspecified, the root directory will be used.
- **report_id** [integer] The ID of the associated report.
- **current_deployment** [dict::]

> > – **deployment_id** [integer] The ID for this deployment.
> >
> > – **user_id** [integer] The ID of the owner.
> >
> > – **host** [string] Domain of the deployment.
> >
> > – **name** [string] Name of the deployment.
> >
> > – **docker_image_name** [string] The name of the docker image to pull from DockerHub.
> >
> > – **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
> >
> > – **display_url** [string] A signed URL for viewing the deployed item.
> >
> > – **instance_type** [string] The EC2 instance type requested for the deployment.

- **memory** [integer] The memory allocated to the deployment, in MB.

- **cpu** [integer] The cpu allocated to the deployment, in millicores.

- **state** [string] The state of the deployment.

- **state_message** [string] A detailed description of the state.

- **max_memory_usage** [number/float] If the deployment has finished, the maximum amount of memory used during the deployment, in MB.

- **max_cpu_usage** [number/float] If the deployment has finished, the maximum amount of cpu used during the deployment, in millicores.

- created_at : string/time

- updated_at : string/time

- **service_id** [integer] The ID of owning Service
- **current_url** [string] The URL that the service is hosted at.
- **environment_variables** [dict] Environment Variables to be passed into the Service.
- **notifications** [dict::]

- **failure_email_addresses** [list] Addresses to notify by e-mail when the service fails.

- **failure_on** [boolean] If failure email notifications are on
- **partition_label** [string] The partition label used to run this object. Only settable with custom_partitions feature flag. Beware attribute may break or change in the future.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.

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

civis.response.Response

- **id** [integer] The ID for this Service.
- **name** [string] The name of this Service.
- **description** [string] The description of this Service.
- **user** [dict::]

- **id** [integer] The ID of this user.

- **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **type** [string] The type of this Service

- **docker_image_name** [string] The name of the docker image to pull from DockerHub.
- **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).
- **schedule** [dict::]

    – **runtime_plan** [string] Only affects the service when deployed. On Demand means that the service will be turned on when viewed and automatically turned off after periods of inactivity. Specific Times means the service will be on when scheduled. Always On means the deployed service will always be on.

    – **recurrences** [list::] List of day-hour combinations this item is scheduled for - scheduled_days : list

        Days it is scheduled on, based on numeric value starting at 0 for Sunday

        ∗ **scheduled_hours** [list] Hours it is scheduled on

- time_zone : string
- **replicas** [integer] The number of Service replicas to deploy. When maxReplicas is set, this field defines the minimum number of replicas to deploy.
- **max_replicas** [integer] The maximum number of Service replicas to deploy. Defining this field enables autoscaling.
- **instance_type** [string] The EC2 instance type to deploy to.
- **memory** [integer] The amount of memory allocated to each replica of the Service.
- **cpu** [integer] The amount of cpu allocated to each replica of the the Service.
- created_at : string/time
- updated_at : string/time
- **credentials** [list] A list of credential IDs to pass to the Service.
- **permission_set_id** [integer] The ID of the associated permission set, if any.
- **git_repo_url** [string] The url for the git repo where the Service code lives.
- **git_repo_ref** [string] The git reference to use when pulling code from the repo.
- **git_path_dir** [string] The path to the Service code within the git repo. If unspecified, the root directory will be used.
- **report_id** [integer] The ID of the associated report.
- **current_deployment** [dict::]

    – **deployment_id** [integer] The ID for this deployment.

    – **user_id** [integer] The ID of the owner.

    – **host** [string] Domain of the deployment.

    – **name** [string] Name of the deployment.

    – **docker_image_name** [string] The name of the docker image to pull from DockerHub.

    – **docker_image_tag** [string] The tag of the docker image to pull from DockerHub (default: latest).

> > > – **display_url** [string] A signed URL for viewing the deployed
> > > item.
> > >
> > > – **instance_type** [string] The EC2 instance type requested for
> > > the deployment.
> > >
> > > – **memory** [integer] The memory allocated to the deployment,
> > > in MB.
> > >
> > > – **cpu** [integer] The cpu allocated to the deployment, in mil-
> > > licores.
> > >
> > > – **state** [string] The state of the deployment.
> > >
> > > – **state_message** [string] A detailed description of the state.
> > >
> > > – **max_memory_usage** [number/float] If the deployment has
> > > finished, the maximum amount of memory used during
> > > the deployment, in MB.
> > >
> > > – **max_cpu_usage** [number/float] If the deployment has fin-
> > > ished, the maximum amount of cpu used during the de-
> > > ployment, in millicores.
> > >
> > > – created_at : string/time
> > >
> > > – updated_at : string/time
> > >
> > > – **service_id** [integer] The ID of owning Service
> >
> > • **current_url** [string] The URL that the service is hosted at.
> > • **environment_variables** [dict] Environment Variables to be passed into
> > the Service.
> > • **notifications** [dict::]
> >
> > > – **failure_email_addresses** [list] Addresses to notify by e-mail
> > > when the service fails.
> > >
> > > – **failure_on** [boolean] If failure email notifications are on
> >
> > • **partition_label** [string] The partition label used to run this object. Only
> > settable with custom_partitions feature flag. Beware attribute may
> > break or change in the future.
> > • **my_permission_level** [string] Your permission level on the object. One
> > of "read", "write", or "manage".
> > • **archived** [string] The archival status of the requested item(s).
> > • **hidden** [boolean] The hidden status of the item.

> **put_projects**(*id*, *project_id*)
>
> > Add a Service to a project
> >
> > > **Parameters**
> > >
> > > > **id** [integer] The ID of the Service.
> > > > **project_id** [integer] The ID of the project.
> > >
> > > **Returns**
> > >
> > > > **None** Response code 204: success

> **put_shares_groups**(*id*, *group_ids*, *permission_level*, ***, *share_email_body='DEFAULT'*,
> *send_shared_email='DEFAULT'*)
>
> > Set the permissions groups has on this object
> >
> > > **Parameters**
> > >
> > > > **id** [integer] The ID of the resource that is shared.
> > > > **group_ids** [list] An array of one or more group IDs.
> > > > **permission_level** [string] Options are: "read", "write", or "manage".

> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> > **Returns**
> > > *civis.response.Response*
> > > > - **readers** [dict::]
> > > > > - **users** [list::]
> > > > > > * id : integer
> > > > > > * name : string
> > > > > - **groups** [list::]
> > > > > > * id : integer
> > > > > > * name : string
> > > > - **writers** [dict::]
> > > > > - **users** [list::]
> > > > > > * id : integer
> > > > > > * name : string
> > > > > - **groups** [list::]
> > > > > > * id : integer
> > > > > > * name : string
> > > > - **owners** [dict::]
> > > > > - **users** [list::]
> > > > > > * id : integer
> > > > > > * name : string
> > > > > - **groups** [list::]
> > > > > > * id : integer
> > > > > > * name : string
> > > > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_ids** [list] An array of one or more user IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> > **Returns**
> > > *civis.response.Response*
> > > > - **readers** [dict::]
> > > > > - **users** [list::]
> > > > > > * id : integer

> > > > > > > > ∗ name : string

> > > > > > > – **groups** [list::]

> > > > > > > > ∗ id : integer

> > > > > > > > ∗ name : string

> > > > > > • **writers** [dict::]

> > > > > > > – **users** [list::]

> > > > > > > > ∗ id : integer

> > > > > > > > ∗ name : string

> > > > > > > – **groups** [list::]

> > > > > > > > ∗ id : integer

> > > > > > > > ∗ name : string

> > > > > > • **owners** [dict::]

> > > > > > > – **users** [list::]

> > > > > > > > ∗ id : integer

> > > > > > > > ∗ name : string

> > > > > > > – **groups** [list::]

> > > > > > > > ∗ id : integer

> > > > > > > > ∗ name : string

> > > > > > • **total_user_shares** [integer] For owners, the number of total users shared.
> > > > > > For writers and readers, the number of visible users shared.
> > > > > > • **total_group_shares** [integer] For owners, the number of total groups
> > > > > > shared. For writers and readers, the number of visible groups shared.

put_transfer(*id*, *user_id*, *include_dependencies*, ***, *email_body='DEFAULT'*, *send_email='DEFAULT'*)
> Transfer ownership of this object to another user
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_id** [integer] ID of target user
> > > **include_dependencies** [boolean] Whether or not to give manage permissions on all
> > > > dependencies
> > > **email_body** [string, optional] Custom body text for e-mail sent on transfer.
> > > **send_email** [boolean, optional] Send email to the target user of the transfer?
> > **Returns**
> > [*civis.response.Response*](#)
> > > > • **dependencies** [list::] Dependent objects for this object - object_type :
> > > > > string

> > > > > > Dependent object type

> > > > > – **fco_type** [string] Human readable dependent object type

> > > > > – **id** [integer] Dependent object ID

> > > > > – **name** [string] Dependent object name, or nil if the requesting
> > > > > > user cannot read this object

> > > > > – **permission_level** [string] Permission level of target user (not
> > > > > > user's groups) for dependent object, or null if no target
> > > > > > user

> > **– shared** [boolean] Whether dependent object was successfully
> > shared with target user

Storage_Hosts

class Storage_Hosts(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.storage_hosts.list(...)
```

Methods

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 a storage host
list()	List the storage hosts
list_dependencies(id, *[, user_id])	List dependent objects for this object
list_shares(id)	List users and groups permissioned on this object
patch(id, *[, name, provider, bucket, ...])	Update some attributes of this storage host
post(provider, bucket, name, *[, s3_options])	Create a new storage host
put(id, name, provider, bucket, *[, s3_options])	Replace all attributes of this storage host
put_shares_groups(id, group_ids, ...[, ...])	Set the permissions groups has on this object
put_shares_users(id, user_ids, ...[, ...])	Set the permissions users have on this object
put_transfer(id, user_id, ...[, email_body, ...])	Transfer ownership of this object to another user

———

delete_shares_groups(*id*, *group_id*)
> Revoke the permissions a group has on this object
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **group_id** [integer] The 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] The ID of the resource that is shared.
> > > **user_id** [integer] The ID of the user.
> > **Returns**
> > > **None** Response code 204: success

get(*id*)
> Get a storage host
> > **Parameters**

> > > **id** [integer] The ID of the storage host.

> > **Returns**
> > > *civis.response.Response*
> > > > - **id** [integer] The ID of the storage host.
> > > > - **owner** [dict::]
> > > > > - **id** [integer] The ID of this user.
> > > > > - **name** [string] This user's name.
> > > > > - **username** [string] This user's username.
> > > > > - **initials** [string] This user's initials.
> > > > > - **online** [boolean] Whether this user is online.
> > > > - **name** [string] The human readable name for the storage host.
> > > > - **provider** [string] The storage provider.One of: s3.
> > > > - **bucket** [string] The bucket for this storage host.
> > > > - **s3_options** [dict::]
> > > > > - **region** [string] The region for this storage host (ex. "us-east-1")

list()
> List the storage hosts
> > **Returns**
> > > *civis.response.Response*
> > > > - **id** [integer] The ID of the storage host.
> > > > - **owner** [dict::]
> > > > > - **id** [integer] The ID of this user.
> > > > > - **name** [string] This user's name.
> > > > > - **username** [string] This user's username.
> > > > > - **initials** [string] This user's initials.
> > > > > - **online** [boolean] Whether this user is online.
> > > > - **name** [string] The human readable name for the storage host.
> > > > - **provider** [string] The storage provider.One of: s3.
> > > > - **bucket** [string] The bucket for this storage host.
> > > > - **s3_options** [dict::]
> > > > > - **region** [string] The region for this storage host (ex. "us-east-1")

list_dependencies(*id*, ***, *user_id='DEFAULT'*)
> List dependent objects for this object
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_id** [integer, optional] ID of target user
> > **Returns**
> > > *civis.response.Response*
> > > > - **object_type** [string] Dependent object type
> > > > - **fco_type** [string] Human readable dependent object type
> > > > - **id** [integer] Dependent object ID
> > > > - **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > > > - **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user

> • **shareable** [boolean] Whether or not the requesting user can share this object.

list_shares(*id*)

> List users and groups permissioned on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **readers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **writers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **owners** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > >
> > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch(*id*, *, *name='DEFAULT'*, *provider='DEFAULT'*, *bucket='DEFAULT'*, *s3_options='DEFAULT'*)

> Update some attributes of this storage host
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the storage host.
> > >
> > > **name** [string, optional] The human readable name for the storage host.
> > >
> > > **provider** [string, optional] The storage provider.One of: s3.
> > >
> > > **bucket** [string, optional] The bucket for this storage host.
> > >
> > > **s3_options** [dict, optional::]
> > >
> > > > • **region** [string] The region for this storage host (ex. "us-east-1")
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)

- **id** [integer] The ID of the storage host.
- **owner** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **name** [string] The human readable name for the storage host.
- **provider** [string] The storage provider.One of: s3.
- **bucket** [string] The bucket for this storage host.
- **s3_options** [dict::]

  - **region** [string] The region for this storage host (ex. "us-east-1")

post(*provider*, *bucket*, *name*, *, *s3_options='DEFAULT'*)

  Create a new storage host

  Parameters

  provider [string] The storage provider.One of: s3.
  bucket [string] The bucket for this storage host.
  name [string] The human readable name for the storage host.
  s3_options [dict, optional::]

  - **region** [string] The region for this storage host (ex. "us-east-1")

  Returns

  [`civis.response.Response`](#)

  - **id** [integer] The ID of the storage host.
  - **owner** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
  - **name** [string] The human readable name for the storage host.
  - **provider** [string] The storage provider.One of: s3.
  - **bucket** [string] The bucket for this storage host.
  - **s3_options** [dict::]

    - **region** [string] The region for this storage host (ex. "us-east-1")

put(*id*, *name*, *provider*, *bucket*, *, *s3_options='DEFAULT'*)

  Replace all attributes of this storage host

  Parameters

  id [integer] The ID of the storage host.
  name [string] The human readable name for the storage host.
  provider [string] The storage provider.One of: s3.
  bucket [string] The bucket for this storage host.
  s3_options [dict, optional::]

  - **region** [string] The region for this storage host (ex. "us-east-1")

  Returns

  [`civis.response.Response`](#)

- **id** [integer] The ID of the storage host.
- **owner** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.
- **name** [string] The human readable name for the storage host.
- **provider** [string] The storage provider.One of: s3.
- **bucket** [string] The bucket for this storage host.
- **s3_options** [dict::]

  – **region** [string] The region for this storage host (ex. "us-east-1")

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

Set the permissions groups has on this object

   Parameters

   id [integer] The ID of the resource that is shared.

   group_ids [list] An array of one or more group IDs.

   permission_level [string] Options are: "read", "write", or "manage".

   share_email_body [string, optional] Custom body text for e-mail sent on a share.

   send_shared_email [boolean, optional] Send email to the recipients of a share.

   Returns

   civis.response.Response

- **readers** [dict::]

  – **users** [list::]

     * id : integer

     * name : string

  – **groups** [list::]

     * id : integer

     * name : string
- **writers** [dict::]

  – **users** [list::]

     * id : integer

     * name : string

  – **groups** [list::]

     * id : integer

     * name : string
- **owners** [dict::]

  – **users** [list::]

     * id : integer

     * name : string

> > > > > > – **groups** [list::]
> > > > > >
> > > > > > > > * id : integer
> > > > > > > >
> > > > > > > > * name : string
> > > > > >
> > > > > > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > > > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_ids** [list] An array of one or more user IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > [*civis.response.Response*](#)
> >
> > > > - **readers** [dict::]
> > > >
> > > > > > – **users** [list::]
> > > > > >
> > > > > > > > * id : integer
> > > > > > > >
> > > > > > > > * name : string
> > > > > >
> > > > > > – **groups** [list::]
> > > > > >
> > > > > > > > * id : integer
> > > > > > > >
> > > > > > > > * name : string
> > > >
> > > > - **writers** [dict::]
> > > >
> > > > > > – **users** [list::]
> > > > > >
> > > > > > > > * id : integer
> > > > > > > >
> > > > > > > > * name : string
> > > > > >
> > > > > > – **groups** [list::]
> > > > > >
> > > > > > > > * id : integer
> > > > > > > >
> > > > > > > > * name : string
> > > >
> > > > - **owners** [dict::]
> > > >
> > > > > > – **users** [list::]
> > > > > >
> > > > > > > > * id : integer
> > > > > > > >
> > > > > > > > * name : string
> > > > > >
> > > > > > – **groups** [list::]
> > > > > >
> > > > > > > > * id : integer
> > > > > > > >
> > > > > > > > * name : string
> > > >
> > > > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_transfer(*id*, *user_id*, *include_dependencies*, *, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

> Transfer ownership of this object to another user
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] ID of target user
> > >
> > > **include_dependencies** [boolean] Whether or not to give manage permissions on all dependencies
> > >
> > > **email_body** [string, optional] Custom body text for e-mail sent on transfer.
> > >
> > > **send_email** [boolean, optional] Send email to the target user of the transfer?
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > • **dependencies** [list::] Dependent objects for this object - object_type : string
> > > >
> > > > > Dependent object type
> > > >
> > > > > – **fco_type** [string] Human readable dependent object type
> > > > >
> > > > > – **id** [integer] Dependent object ID
> > > > >
> > > > > – **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > > > >
> > > > > – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > > > >
> > > > > – **shared** [boolean] Whether dependent object was successfully shared with target user

Table_Tags

class **Table_Tags**(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.table_tags.list(...)
```

Methods

delete(id)	Delete a Table Tag
get(id)	Get a Table Tag
list(*[, name, limit, page_num, order, ...])	List Table Tags
post(name)	Create a Table Tag

———

delete(*id*)

> Delete a Table Tag

> > > > **Parameters**
> > > > > **id** [integer]
> > > > **Returns**
> > > > > **None** Response code 204: success

> > **get**(*id*)
> > > Get a Table Tag
> > > > **Parameters**
> > > > > **id** [integer]
> > > > **Returns**
> > > > > [*civis.response.Response*](#)
> > > > > > - **id** [integer] Table Tag ID
> > > > > > - **name** [string] Table Tag Name
> > > > > > - **created_at** [string/date-time] The date the tag was created.
> > > > > > - **updated_at** [string/date-time] The date the tag was recently updated on.
> > > > > > - **table_count** [integer] The total number of tables associated with the tag.
> > > > > > - **user** [dict::]
> > > > > > > - **id** [integer] The ID of this user.
> > > > > > > - **name** [string] This user's name.
> > > > > > > - **username** [string] This user's username.
> > > > > > > - **initials** [string] This user's initials.
> > > > > > > - **online** [boolean] Whether this user is online.

> > **list**(***, *name='DEFAULT'*, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)
> > > List Table Tags
> > > > **Parameters**
> > > > > **name** [string, optional] Name of the tag. If it is provided, the results will be filtered by name
> > > > > **limit** [integer, optional] Number of results to return. Defaults to 50. Maximum allowed is 1000.
> > > > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > > > > **order** [string, optional] The field on which to order the result set. Defaults to name. Must be one of: name, user, table_count.
> > > > > **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**
> > > > > [*civis.response.PaginatedResponse*](#)
> > > > > > - **id** [integer] Table Tag ID
> > > > > > - **name** [string] Table Tag Name
> > > > > > - **table_count** [integer] The total number of tables associated with the tag.
> > > > > > - **user** [dict::]
> > > > > > > - **id** [integer] The ID of this user.
> > > > > > > - **name** [string] This user's name.
> > > > > > > - **username** [string] This user's username.
> > > > > > > - **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

post(*name*)

    Create a Table Tag

        Parameters

            name [string] Table Tag Name

        Returns

            civis.response.Response

- **id** [integer] Table Tag ID
- **name** [string] Table Tag Name
- **created_at** [string/date-time] The date the tag was created.
- **updated_at** [string/date-time] The date the tag was recently updated on.
- **table_count** [integer] The total number of tables associated with the tag.
- **user** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

Tables

class Tables(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.tables.post_enhancements_geocodings(...)
```

Methods

`delete_projects`(id, project_id)	Remove a Table from a project
`delete_tags`(id, table_tag_id)	Add a tag to a table
`get`(id)	Show basic table info
`get_enhancements_cass_ncoa`(id, source_table_id)	Deprecation warning! :2: (WARNING/2) Title underline too short. Deprecation warning! ------------------ Warning: The tables/:source_table_id/enhancements/cass-ncoa/:id endpoint is deprecated and will be removed after January 1, 2021. View the status of a CASS / NCOA table enhancement

continues on next page

`get_enhancements_geocodings`(id, source_table_id)	Deprecation warning! :2: (WARNING/2) Title underline too short. Deprecation warning! ------------------ Warning: The tables/:source_table_id/enhancements/geocodings/:id endpoint is deprecated and will be removed after January 1, 2021. View the status of a geocoding table enhancement
`list`(*[, database_id, schema, name, search, ...])	List tables
`list_columns`(id, *[, name, limit, page_num, ...])	List columns in the specified table
`list_projects`(id, *[, hidden])	List the projects a Table belongs to
`patch`(id, *[, ontology_mapping, ...])	Update a table
`post_enhancements_cass_ncoa`(source_table_id, *)	Deprecation warning! :2: (WARNING/2) Title underline too short. Deprecation warning! ------------------ Warning: The tables/:source_table_id/enhancements/cass-ncoa endpoint is deprecated and will be removed after January 1, 2021. Standardize addresses in a table
`post_enhancements_geocodings`(source_table_id)	Deprecation warning! :2: (WARNING/2) Title underline too short. Deprecation warning! ------------------ Warning: The tables/:source_table_id/enhancements/geocodings endpoint is deprecated and will be removed after January 1, 2021. Geocode a table
`post_refresh`(id)	Deprecation warning! :2: (WARNING/2) Title underline too short. Deprecation warning! ------------------ Warning: The tables/:id/refresh endpoint is deprecated. Please use tables/scan from now on. Request a refresh for column and table statistics
`post_scan`(database_id, schema, table_name, *)	Creates and enqueues a single table scanner job on a new table
`put_projects`(id, project_id)	Add a Table to a project
`put_tags`(id, table_tag_id)	Add a tag to a table

---

delete_projects(*id*, *project_id*)

Remove a Table from a project

> **Parameters**
>> **id** [integer] The ID of the Table.
>> **project_id** [integer] The ID of the project.
>
> **Returns**
>> **None** Response code 204: success

delete_tags(*id*, *table_tag_id*)

Add a tag to a table

> **Parameters**
>> **id** [integer] The ID of the table.
>> **table_tag_id** [integer] The ID of the tag.
>
> **Returns**

None Response code 200: success

get(*id*)

Show basic table info

Parameters

id [integer]

Returns

civis.response.Response

- **id** [integer] The ID of the table.
- **database_id** [integer] The ID of the database.
- **schema** [string] The name of the schema containing the table.
- **name** [string] Name of the table.
- **description** [string] The description of the table, as specified by the table owner
- **is_view** [boolean] True if this table represents a view. False if it represents a regular table.
- **row_count** [integer] The number of rows in the table.
- **column_count** [integer] The number of columns in the table.
- **size_mb** [number/float] The size of the table in megabytes.
- **owner** [string] The database username of the table's owner.
- **distkey** [string] The column used as the Amazon Redshift distkey.
- **sortkeys** [string] The column used as the Amazon Redshift sortkey.
- **refresh_status** [string] How up-to-date the table's statistics on row counts, null counts, distinct counts, and values distributions are. One of: refreshing, stale, or current.
- **last_refresh** [string/date-time] The time of the last statistics refresh.
- **data_updated_at** [string/date-time] The last time that Civis Platform captured a change in this table.Only applicable for Redshift tables; please see the Civis help desk for more info.
- **schema_updated_at** [string/date-time] The last time that Civis Platform captured a change to the table attributes/structure.Only applicable for Redshift tables; please see the Civis help desk for more info.
- **refresh_id** [string] The ID of the most recent statistics refresh.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **primary_keys** [list] The primary keys for this table.
- **last_modified_keys** [list] The columns indicating an entry's modification status for this table.
- **table_tags** [list::] The table tags associated with this table. - id : integer

    Table Tag ID

    - **name** [string] Table Tag Name
- **ontology_mapping** [dict] The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.
- **columns** [list::]

- **name** [string] Name of the column.

- **civis_data_type** [string] The generic data type of the column (ex. "string"). Since this is database-agnostic, it may be helpful when loading data to R/Python.

- **sql_type** [string] The database-specific SQL type of the column (ex. "varchar(30)").

- **sample_values** [list] A sample of values from the column.

- **encoding** [string] The compression encoding for this columnSee: http://docs.aws.amazon .com/redshift/latest/dg/c_Compression_encodings.html

- **description** [string] The description of the column, as specified by the table owner

- **order** [integer] Relative position of the column in the table.

- **min_value** [string] Smallest value in the column.

- **max_value** [string] Largest value in the column.

- **avg_value** [number/float] This parameter is deprecated.

- **stddev** [number/float] This parameter is deprecated.

- **value_distribution_percent** [dict] A mapping between each value in the column and the percentage of rows with that value.Only present for tables with fewer than approximately 25,000,000 rows and for columns with fewer than twenty distinct values.

- **coverage_count** [integer] Number of non-null values in the column.

- **null_count** [integer] Number of null values in the column.

- **possible_dependent_variable_types** [list] Possible dependent variable types the column may be used to model. Null if it may not be used as a dependent variable.

- **useable_as_independent_variable** [boolean] Whether the column may be used as an independent variable to train a model.

- **useable_as_primary_key** [boolean] Whether the column may be used as an primary key to identify table rows.

- **value_distribution** [dict] An object mapping distinct values in the column to the number of times they appear in the column

- **distinct_count** [integer] Number of distinct values in the column. NULL values are counted and treated as a single distinct value.

- **joins** [list::]

    - id : integer

    - left_table_id : integer

    - left_identifier : string

- **–** right_table_id : integer

- **–** right_identifier : string

- **–** on : string

- **–** left_join : boolean

- **–** created_at : string/time

- **–** updated_at : string/time
- multipart_key : list
- **enhancements** [list::]

- **–** type : string

- **–** created_at : string/time

- **–** updated_at : string/time

- **–** join_id : integer
- view_def : string
- table_def : string
- **outgoing_table_matches** [list::]

- **–** **source_table_id** [integer] Source table

- **–** **target_type** [string] Target type

- **–** **target_id** [integer] Target ID

- **–** **target** [dict::]

    - ∗ name : string

- **–** **job** [dict::]

    - ∗ id : integer

    - ∗ name : string

    - ∗ type : string

    - ∗ from_template_id : integer

    - ∗ **state** [string] Whether the job is idle, queued, running, cancelled, or failed.

    - ∗ created_at : string/date-time

    - ∗ updated_at : string/date-time

    - ∗ **runs** [list::] Information about the most recent runs of the job. - id : integer - state : string - created_at : string/time

        The time that the run was queued.

        - · **started_at** [string/time] The time that the run started.

        - · **finished_at** [string/time] The time that the run completed.

        - · **error** [string] The error message for this run, if present.

        ∗ **last_run** [dict::]

            · id : integer

            · state : string

            · **created_at** [string/time] The time that the run was queued.

            · **started_at** [string/time] The time that the run started.

            · **finished_at** [string/time] The time that the run completed.

            · **error** [string] The error message for this run, if present.

        ∗ **hidden** [boolean] The hidden status of the item.

        ∗ **match_options** [dict::]

            · max_matches : integer

            · threshold : string

get_enhancements_cass_ncoa(*id*, *source_table_id*)

    Warning: The tables/:source_table_id/enhancements/cass-ncoa/:id endpoint is deprecated and will be removed after January 1, 2021. 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

        civis.response.Response

            • **id** [integer] The ID of the enhancement.
            • **source_table_id** [integer] The ID of the table that was enhanced.
            • **state** [string] The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
            • **enhanced_table_schema** [string] The schema name of the table created by the enhancement.
            • **enhanced_table_name** [string] The name of the table created by the enhancement.
            • **perform_ncoa** [boolean] Whether to update addresses for records matching the National Change of Address (NCOA) database.
            • **ncoa_credential_id** [integer] Credential to use when performing NCOA updates. Required if 'performNcoa' is true.
            • **output_level** [string] The set of fields persisted by a CASS or NCOA enhancement.For CASS enhancements, one of 'cass' or 'all.'For NCOA enhancements, one of 'cass', 'ncoa' , 'coalesced' or 'all'.By default, all fields will be returned.

get_enhancements_geocodings(*id*, *source_table_id*)

    Warning: The tables/:source_table_id/enhancements/geocodings/:id endpoint is deprecated and will be removed after January 1, 2021. 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

        civis.response.Response

            • **id** [integer] The ID of the enhancement.

> - **source_table_id** [integer] The ID of the table that was enhanced.
> - **state** [string] The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
> - **enhanced_table_schema** [string] The schema name of the table created by the enhancement.
> - **enhanced_table_name** [string] The name of the table created by the enhancement.

list(*, *database_id='DEFAULT'*, *schema='DEFAULT'*, *name='DEFAULT'*, *search='DEFAULT'*, *table_tag_ids='DEFAULT'*, *credential_id='DEFAULT'*, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)

List tables

> **Parameters**
>> **database_id** [integer, optional] The ID of the database.
>>
>> **schema** [string, optional] If specified, will be used to filter the tables returned. Substring matching is supported with "%" and "*" wildcards (e.g., "schema=%census%" will return both "client_census.table" and "census_2010.table").
>>
>> **name** [string, optional] If specified, will be used to filter the tables returned. Substring matching is supported with "%" and "*" wildcards (e.g., "name=%table%" will return both "table1" and "my table").
>>
>> **search** [string, optional] If specified, will be used to filter the tables returned. Will search across schema and name (in the full form schema.name) and will return any full name containing the search string.
>>
>> **table_tag_ids** [array, optional] If specified, will be used to filter the tables returned. Will search across Table Tags and will return any tables that have one of the matching Table Tags.
>>
>> **credential_id** [integer, optional] If specified, will be used instead of the default credential to filter the tables returned.
>>
>> **limit** [integer, optional] Number of results to return. Defaults to 50. Maximum allowed is 1000.
>>
>> **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
>>
>> **order** [string, optional] The field on which to order the result set. Defaults to schema. Must be one of: schema, name, search, table_tag_ids, credential_id.
>>
>> **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**
>
> *civis.response.PaginatedResponse*
>> - **id** [integer] The ID of the table.
>> - **database_id** [integer] The ID of the database.
>> - **schema** [string] The name of the schema containing the table.
>> - **name** [string] Name of the table.
>> - **description** [string] The description of the table, as specified by the table owner
>> - **is_view** [boolean] True if this table represents a view. False if it represents a regular table.
>> - **row_count** [integer] The number of rows in the table.
>> - **column_count** [integer] The number of columns in the table.
>> - **size_mb** [number/float] The size of the table in megabytes.
>> - **owner** [string] The database username of the table's owner.

- **distkey** [string] The column used as the Amazon Redshift distkey.
- **sortkeys** [string] The column used as the Amazon Redshift sortkey.
- **refresh_status** [string] How up-to-date the table's statistics on row counts, null counts, distinct counts, and values distributions are. One of: refreshing, stale, or current.
- **last_refresh** [string/date-time] The time of the last statistics refresh.
- **refresh_id** [string] The ID of the most recent statistics refresh.
- **last_run** [dict::]

    – id : integer

    – state : string

    – **created_at** [string/time] The time that the run was queued.

    – **started_at** [string/time] The time that the run started.

    – **finished_at** [string/time] The time that the run completed.

    – **error** [string] The error message for this run, if present.
- **table_tags** [list::] The table tags associated with this table. - id : integer

    Table Tag ID

    – **name** [string] Table Tag Name

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

List columns in the specified table

> **Parameters**
> > **id** [integer]
> > **name** [string, optional] Search for columns with the given name, within the specified table.
> > **limit** [integer, optional] Number of results to return. Defaults to its maximum of 50.
> > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > **order** [string, optional] The field on which to order the result set. Defaults to name. Must be one of: name, order.
> > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to asc.
> > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
>
> **Returns**
> > *civis.response.PaginatedResponse*
> > > - **name** [string] Name of the column.
> > > - **civis_data_type** [string] The generic data type of the column (ex. "string"). Since this is database-agnostic, it may be helpful when loading data to R/Python.
> > > - **sql_type** [string] The database-specific SQL type of the column (ex. "varchar(30)").
> > > - **sample_values** [list] A sample of values from the column.
> > > - **encoding** [string] The compression encoding for this columnSee: http://docs.aws.amazon.com /redshift/latest/dg/c_Compression_encodings.html
> > > - **description** [string] The description of the column, as specified by the table owner
> > > - **order** [integer] Relative position of the column in the table.

- **min_value** [string] Smallest value in the column.
- **max_value** [string] Largest value in the column.
- **avg_value** [number/float] This parameter is deprecated.
- **stddev** [number/float] This parameter is deprecated.
- **value_distribution_percent** [dict] A mapping between each value in the column and the percentage of rows with that value.Only present for tables with fewer than approximately 25,000,000 rows and for columns with fewer than twenty distinct values.
- **coverage_count** [integer] Number of non-null values in the column.
- **null_count** [integer] Number of null values in the column.
- **possible_dependent_variable_types** [list] Possible dependent variable types the column may be used to model. Null if it may not be used as a dependent variable.
- **useable_as_independent_variable** [boolean] Whether the column may be used as an independent variable to train a model.
- **useable_as_primary_key** [boolean] Whether the column may be used as an primary key to identify table rows.
- **value_distribution** [dict] An object mapping distinct values in the column to the number of times they appear in the column
- **distinct_count** [integer] Number of distinct values in the column. NULL values are counted and treated as a single distinct value.

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

    List the projects a Table belongs to

        Parameters

            id [integer] The ID of the Table.

            hidden [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.

        Returns

        civis.response.Response

- **id** [integer] The ID for this project.
- **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- **name** [string] The name of this project.
- **description** [string] A description of the project.
- **users** [list::] Users who can see the project. - id : integer

    The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.
- auto_share : boolean
- created_at : string/time
- updated_at : string/time

> > > > • **archived** [string] The archival status of the requested item(s).

patch(*id*, ***, *ontology_mapping='DEFAULT'*, *description='DEFAULT'*, *primary_keys='DEFAULT'*,
      last_modified_keys='DEFAULT')

> Update a table
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the table.
> > >
> > > **ontology_mapping** [dict, optional] The ontology-key to column-name mapping. See
> > > /ontology for the list of valid ontology keys.
> > >
> > > **description** [string, optional] The user-defined description of the table.
> > >
> > > **primary_keys** [list, optional] A list of column(s) which together uniquely identify a
> > > row in the data.These columns must not contain NULL values.
> > >
> > > **last_modified_keys** [list, optional] The columns indicating when a row was last mod-
> > > ified.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **id** [integer] The ID of the table.
> > > > • **database_id** [integer] The ID of the database.
> > > > • **schema** [string] The name of the schema containing the table.
> > > > • **name** [string] Name of the table.
> > > > • **description** [string] The description of the table, as specified by the table
> > > >   owner
> > > > • **is_view** [boolean] True if this table represents a view. False if it represents
> > > >   a regular table.
> > > > • **row_count** [integer] The number of rows in the table.
> > > > • **column_count** [integer] The number of columns in the table.
> > > > • **size_mb** [number/float] The size of the table in megabytes.
> > > > • **owner** [string] The database username of the table's owner.
> > > > • **distkey** [string] The column used as the Amazon Redshift distkey.
> > > > • **sortkeys** [string] The column used as the Amazon Redshift sortkey.
> > > > • **refresh_status** [string] How up-to-date the table's statistics on row counts,
> > > >   null counts, distinct counts, and values distributions are. One of:
> > > >   refreshing, stale, or current.
> > > > • **last_refresh** [string/date-time] The time of the last statistics refresh.
> > > > • **data_updated_at** [string/date-time] The last time that Civis Platform cap-
> > > >   tured a change in this table.Only applicable for Redshift tables;
> > > >   please see the Civis help desk for more info.
> > > > • **schema_updated_at** [string/date-time] The last time that Civis Platform
> > > >   captured a change to the table attributes/structure.Only applicable
> > > >   for Redshift tables; please see the Civis help desk for more info.
> > > > • **refresh_id** [string] The ID of the most recent statistics refresh.
> > > > • **last_run** [dict::]
> > > >
> > > > > – id : integer
> > > > >
> > > > > – state : string
> > > > >
> > > > > – **created_at** [string/time] The time that the run was queued.
> > > > >
> > > > > – **started_at** [string/time] The time that the run started.
> > > > >
> > > > > – **finished_at** [string/time] The time that the run completed.
> > > > >
> > > > > – **error** [string] The error message for this run, if present.
> > > > • **primary_keys** [list] The primary keys for this table.
> > > > • **last_modified_keys** [list] The columns indicating an entry's modification
> > > >   status for this table.

- **table_tags** [list::] The table tags associated with this table. - id : integer
  Table Tag ID

  – **name** [string] Table Tag Name

- **ontology_mapping** [dict] The ontology-key to column-name mapping.
  See /ontology for the list of valid ontology keys.

post_enhancements_cass_ncoa(*source_table_id*, ***, *perform_ncoa='DEFAULT'*,
                    ncoa_credential_id='DEFAULT', *output_level='DEFAULT'*)

    Warning: The tables/:source_table_id/enhancements/cass-ncoa endpoint is deprecated and will be removed after January 1, 2021. Standardize addresses in a table

        Parameters

            source_table_id [integer] The ID of the table to be enhanced.

            perform_ncoa [boolean, optional] Whether to update addresses for records matching the National Change of Address (NCOA) database.

            ncoa_credential_id [integer, optional] Credential to use when performing NCOA updates. Required if 'performNcoa' is true.

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

        Returns

        civis.response.Response

- **id** [integer] The ID of the enhancement.
- **source_table_id** [integer] The ID of the table that was enhanced.
- **state** [string] The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **enhanced_table_schema** [string] The schema name of the table created by the enhancement.
- **enhanced_table_name** [string] The name of the table created by the enhancement.
- **perform_ncoa** [boolean] Whether to update addresses for records matching the National Change of Address (NCOA) database.
- **ncoa_credential_id** [integer] Credential to use when performing NCOA updates. Required if 'performNcoa' is true.
- **output_level** [string] The set of fields persisted by a CASS or NCOA enhancement.For CASS enhancements, one of 'cass' or 'all.'For NCOA enhancements, one of 'cass', 'ncoa' , 'coalesced' or 'all'.By default, all fields will be returned.

post_enhancements_geocodings(*source_table_id*)

    Warning: The tables/:source_table_id/enhancements/geocodings endpoint is deprecated and will be removed after January 1, 2021. Geocode a table

        Parameters

            source_table_id [integer] The ID of the table to be enhanced.

        Returns

        civis.response.Response

- **id** [integer] The ID of the enhancement.
- **source_table_id** [integer] The ID of the table that was enhanced.
- **state** [string] The state of the enhancement, one of 'queued' 'running' 'succeeded' 'failed' or 'cancelled'.
- **enhanced_table_schema** [string] The schema name of the table created by the enhancement.

- **enhanced_table_name** [string] The name of the table created by the enhancement.

post_refresh(*id*)

Warning: The tables/:id/refresh endpoint is deprecated. Please use tables/scan from now on. Request a refresh for column and table statistics

Parameters

id [integer]

Returns

civis.response.Response

- **id** [integer] The ID of the table.
- **database_id** [integer] The ID of the database.
- **schema** [string] The name of the schema containing the table.
- **name** [string] Name of the table.
- **description** [string] The description of the table, as specified by the table owner
- **is_view** [boolean] True if this table represents a view. False if it represents a regular table.
- **row_count** [integer] The number of rows in the table.
- **column_count** [integer] The number of columns in the table.
- **size_mb** [number/float] The size of the table in megabytes.
- **owner** [string] The database username of the table's owner.
- **distkey** [string] The column used as the Amazon Redshift distkey.
- **sortkeys** [string] The column used as the Amazon Redshift sortkey.
- **refresh_status** [string] How up-to-date the table's statistics on row counts, null counts, distinct counts, and values distributions are. One of: refreshing, stale, or current.
- **last_refresh** [string/date-time] The time of the last statistics refresh.
- **data_updated_at** [string/date-time] The last time that Civis Platform captured a change in this table.Only applicable for Redshift tables; please see the Civis help desk for more info.
- **schema_updated_at** [string/date-time] The last time that Civis Platform captured a change to the table attributes/structure.Only applicable for Redshift tables; please see the Civis help desk for more info.
- **refresh_id** [string] The ID of the most recent statistics refresh.
- **last_run** [dict::]

    - id : integer

    - state : string

    - **created_at** [string/time] The time that the run was queued.

    - **started_at** [string/time] The time that the run started.

    - **finished_at** [string/time] The time that the run completed.

    - **error** [string] The error message for this run, if present.
- **primary_keys** [list] The primary keys for this table.
- **last_modified_keys** [list] The columns indicating an entry's modification status for this table.
- **table_tags** [list::] The table tags associated with this table. - id : integer

    Table Tag ID

    - **name** [string] Table Tag Name

- **ontology_mapping** [dict] The ontology-key to column-name mapping. See /ontology for the list of valid ontology keys.
- **columns** [list::]

  - **name** [string] Name of the column.

  - **civis_data_type** [string] The generic data type of the column (ex. "string"). Since this is database-agnostic, it may be helpful when loading data to R/Python.

  - **sql_type** [string] The database-specific SQL type of the column (ex. "varchar(30)").

  - **sample_values** [list] A sample of values from the column.

  - **encoding** [string] The compression encoding for this columnSee: http://docs.aws.amazon .com/redshift/latest/dg/c_Compression_encodings.html

  - **description** [string] The description of the column, as specified by the table owner

  - **order** [integer] Relative position of the column in the table.

  - **min_value** [string] Smallest value in the column.

  - **max_value** [string] Largest value in the column.

  - **avg_value** [number/float] This parameter is deprecated.

  - **stddev** [number/float] This parameter is deprecated.

  - **value_distribution_percent** [dict] A mapping between each value in the column and the percentage of rows with that value.Only present for tables with fewer than approximately 25,000,000 rows and for columns with fewer than twenty distinct values.

  - **coverage_count** [integer] Number of non-null values in the column.

  - **null_count** [integer] Number of null values in the column.

  - **possible_dependent_variable_types** [list] Possible dependent variable types the column may be used to model. Null if it may not be used as a dependent variable.

  - **useable_as_independent_variable** [boolean] Whether the column may be used as an independent variable to train a model.

  - **useable_as_primary_key** [boolean] Whether the column may be used as an primary key to identify table rows.

  - **value_distribution** [dict] An object mapping distinct values in the column to the number of times they appear in the column

  - **distinct_count** [integer] Number of distinct values in the column. NULL values are counted and treated as a single distinct value.

- **joins** [list::]

  - id : integer

- – left_table_id : integer

- – left_identifier : string

- – right_table_id : integer

- – right_identifier : string

- – on : string

- – left_join : boolean

- – created_at : string/time

- – updated_at : string/time
- multipart_key : list
- **enhancements** [list::]

  - – type : string

  - – created_at : string/time

  - – updated_at : string/time

  - – join_id : integer
- view_def : string
- table_def : string
- **outgoing_table_matches** [list::]

  - – **source_table_id** [integer] Source table

  - – **target_type** [string] Target type

  - – **target_id** [integer] Target ID

  - – **target** [dict::]

    - ∗ name : string

  - – **job** [dict::]

    - ∗ id : integer

    - ∗ name : string

    - ∗ type : string

    - ∗ from_template_id : integer

    - ∗ **state** [string] Whether the job is idle, queued, running, cancelled, or failed.

    - ∗ created_at : string/date-time

    - ∗ updated_at : string/date-time

    - ∗ **runs** [list::] Information about the most recent runs of the job. - id : integer - state : string - created_at : string/time

      The time that the run was queued.

      - · **started_at** [string/time] The time that the run started.

      - · **finished_at** [string/time] The time that the run completed.

> · **error** [string] The error message for this run, if present.

> * **last_run** [dict::]

>> · id : integer

>> · state : string

>> · **created_at** [string/time] The time that the run was queued.

>> · **started_at** [string/time] The time that the run started.

>> · **finished_at** [string/time] The time that the run completed.

>> · **error** [string] The error message for this run, if present.

> * **hidden** [boolean] The hidden status of the item.

> * **match_options** [dict::]

>> · max_matches : integer

>> · threshold : string

post_scan(*database_id*, *schema*, *table_name*, *, *stats_priority='DEFAULT'*)

> Creates and enqueues a single table scanner job on a new table

>> **Parameters**

>>> **database_id** [integer] The ID of the database.

>>> **schema** [string] The name of the schema containing the table.

>>> **table_name** [string] The name of the table.

>>> **stats_priority** [string, optional] When to sync table statistics. Valid Options are the following. Option: 'flag' means to flag stats for the next scheduled run of a full table scan on the database. Option: 'block' means to block this job on stats syncing. Option: 'queue' means to queue a separate job for syncing stats and do not block this job on the queued job. Defaults to 'flag'

>> **Returns**

>>> *civis.response.Response*

>>>> • **job_id** [integer] The ID of the job created.

>>>> • **run_id** [integer] The ID of the run created.

put_projects(*id*, *project_id*)

> Add a Table to a project

>> **Parameters**

>>> **id** [integer] The ID of the Table.

>>> **project_id** [integer] The ID of the project.

>> **Returns**

>>> **None** Response code 204: success

put_tags(*id*, *table_tag_id*)

> Add a tag to a table

>> **Parameters**

>>> **id** [integer] The ID of the table.

>>> **table_tag_id** [integer] The ID of the tag.

>> **Returns**

>>> *civis.response.Response*

> • **id** [integer] The ID of the table.
> • **table_tag_id** [integer] The ID of the tag.

Templates

class Templates(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.templates.list_reports_shares(...)
```

Methods

delete_reports_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_reports_shares_users(id, user_id)	Revoke the permissions a user has on this object
delete_scripts_projects(id, project_id)	Remove a Script Template from a project
delete_scripts_shares_groups(id, group_id)	Revoke the permissions a group has on this object
delete_scripts_shares_users(id, user_id)	Revoke the permissions a user has on this object
get_reports(id)	Get a Report Template
get_scripts(id)	Get a Script Template
list_reports(*[, hidden, author, category, ...])	List Report Templates
list_reports_dependencies(id, *[, user_id])	List dependent objects for this object
list_reports_shares(id)	List users and groups permissioned on this object
list_scripts(*[, hidden, author, category, ...])	List Script Templates
list_scripts_dependencies(id, *[, user_id])	List dependent objects for this object
list_scripts_projects(id, *[, hidden])	List the projects a Script Template belongs to
list_scripts_shares(id)	List users and groups permissioned on this object
patch_reports(id, *[, name, category, ...])	Update some attributes of this Report Template
patch_scripts(id, *[, name, note, ...])	Update some attributes of this Script Template
post_reports(name, code_body, *[, category, ...])	Create a Report Template
post_reports_review(id, status)	Review a template for security vulnerability and correctness (admin-only)
post_scripts(script_id, name, *[, note, ...])	Create a Script Template
post_scripts_review(id, status)	Review a template for security vulnerability and correctness (admin-only)
put_reports(id, name, code_body, *[, ...])	Replace all attributes of this Report Template
put_reports_shares_groups(id, group_ids, ...)	Set the permissions groups has on this object
put_reports_shares_users(id, user_ids, ...)	Set the permissions users have on this object
put_reports_transfer(id, user_id, ...[, ...])	Transfer ownership of this object to another user
put_scripts(id, name, *[, note, ...])	Replace all attributes of this Script Template
put_scripts_projects(id, project_id)	Add a Script Template to a project
put_scripts_shares_groups(id, group_ids, ...)	Set the permissions groups has on this object
put_scripts_shares_users(id, user_ids, ...)	Set the permissions users have on this object
put_scripts_transfer(id, user_id, ...[, ...])	Transfer ownership of this object to another user

—

delete_reports_shares_groups(*id*, *group_id*)

> Revoke the permissions a group has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **group_id** [integer] The ID of the group.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

delete_reports_shares_users(*id*, *user_id*)

> Revoke the permissions a user has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] The ID of the user.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

delete_scripts_projects(*id*, *project_id*)

> Remove a Script Template from a project
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the Script Template.
> > >
> > > **project_id** [integer] The ID of the project.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

delete_scripts_shares_groups(*id*, *group_id*)

> Revoke the permissions a group has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **group_id** [integer] The ID of the group.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

delete_scripts_shares_users(*id*, *user_id*)

> Revoke the permissions a user has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] The ID of the user.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

get_reports(*id*)

> Get a Report Template
>
> > **Parameters**
> >
> > > **id** [integer]
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > - id : integer
> > > > - **name** [string] The name of the template.
> > > > - **category** [string] The category of this report template. Can be left blank.
> > > >   Acceptable values are: dataset-viz
> > > > - created_at : string/time
> > > > - updated_at : string/time
> > > > - **use_count** [integer] The number of uses of this template.
> > > > - **archived** [boolean] Whether the template has been archived.

- **tech_reviewed** [boolean] Whether this template has been audited by Civis for security vulnerability and correctness.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **auth_code_url** [string] A URL to the template's stored code body.
- **provide_api_key** [boolean] Whether reports based on this template request an API Key from the report viewer.
- **hidden** [boolean] The hidden status of the item.

get_scripts(*id*)

Get a Script Template

    Parameters

        id [integer]

    Returns

        civis.response.Response

- id : integer
- **public** [boolean] If the template is public or not.
- **script_id** [integer] The id of the script that this template uses.
- **script_type** [string] The type of the template's backing script (e.g SQL, Container, Python, R, JavaScript)
- **user_context** [string] The user context of the script that this template uses.
- **params** [list::] A definition of the parameters that this template's backing script accepts in the arguments field. - name : string

    The variable's name as used within your code.

    - **label** [string] The label to present to users when asking them for the value.

    - **description** [string] A short sentence or fragment describing this parameter to the end user.

    - **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

    - **required** [boolean] Whether this param is required.

    - **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

    - **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

    - **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter.

> > > > > Allowed values is an array of hashes of the following
> > > > > format: *{label: 'Import', 'value': 'import'}*

- **name** [string] The name of the template.
- **category** [string] The category of this template.
- **note** [string] A note describing what this template is used for; custom
  scripts created off this template will display this description.
- created_at : string/time
- updated_at : string/time
- **use_count** [integer] The number of uses of this template.
- **ui_report_id** [integer] The id of the report that this template uses.
- **tech_reviewed** [boolean] Whether this template has been audited by Civis
  for security vulnerability and correctness.
- **archived** [boolean] Whether the template has been archived.
- **hidden** [boolean] The hidden status of the item.
- **author** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

list_reports(*, *hidden='DEFAULT'*, *author='DEFAULT'*, *category='DEFAULT'*, *limit='DEFAULT'*,
  page_num='DEFAULT', *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)

> List Report Templates
>
> > **Parameters**
> >
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to
> > >   false, returning non-hidden items.
> > >
> > > **author** [string, optional] If specified, return items from any of these authors. It accepts
> > >   a comma- separated list of user IDs.
> > >
> > > **category** [string, optional] A category to filter results by, one of: dataset-viz
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to 50. Maximum allowed
> > >   is 1000.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first
> > >   page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to name.
> > >   Must be one of: name, updated_at, created_at.
> > >
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc
> > >   (descending) defaulting to asc.
> > >
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use
> > >   when more results than the maximum allowed by limit are needed. When True,
> > >   limit and page_num are ignored. Defaults to False.
> >
> > **Returns**
> >
> > > *civis.response.PaginatedResponse*
> > >
> > > - id : integer
> > > - **name** [string] The name of the template.
> > > - **category** [string] The category of this report template. Can be left blank.
> > >   Acceptable values are: dataset-viz
> > > - created_at : string/time
> > > - updated_at : string/time
> > > - **use_count** [integer] The number of uses of this template.
> > > - **archived** [boolean] Whether the template has been archived.

- **tech_reviewed** [boolean] Whether this template has been audited by Civis
  for security vulnerability and correctness.
- **author** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.

list_reports_dependencies(*id*, ***, *user_id='DEFAULT'*)

List dependent objects for this object

Parameters

id [integer] The ID of the resource that is shared.

user_id [integer, optional] ID of target user

Returns

civis.response.Response

- **object_type** [string] Dependent object type
- **fco_type** [string] Human readable dependent object type
- **id** [integer] Dependent object ID
- **name** [string] Dependent object name, or nil if the requesting user cannot
  read this object
- **permission_level** [string] Permission level of target user (not user's
  groups) for dependent object, or null if no target user
- **shareable** [boolean] Whether or not the requesting user can share this object.

list_reports_shares(*id*)

List users and groups permissioned on this object

Parameters

id [integer] The ID of the resource that is shared.

Returns

civis.response.Response

- **readers** [dict::]

  – **users** [list::]

      ∗ id : integer

      ∗ name : string

  – **groups** [list::]

      ∗ id : integer

      ∗ name : string

- **writers** [dict::]

  – **users** [list::]

      ∗ id : integer

      ∗ name : string

  – **groups** [list::]

      ∗ id : integer

      ∗ name : string

> > > > - **owners** [dict::]
> > > >
> > > > > - **users** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > > >
> > > > > - **groups** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > >
> > > > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

list_scripts(*, *hidden='DEFAULT'*, *author='DEFAULT'*, *category='DEFAULT'*, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)

> List Script Templates
>
> > **Parameters**
> >
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> > >
> > > **author** [string, optional] If specified, return items from any of these authors. It accepts a comma- separated list of user IDs.
> > >
> > > **category** [string, optional] A category to filter results by, one of: import, export, enhancement, model, and script
> > >
> > > **limit** [integer, optional] Number of results to return. Defaults to 50. Maximum allowed is 1000.
> > >
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > >
> > > **order** [string, optional] The field on which to order the result set. Defaults to name. Must be one of: name, updated_at, created_at.
> > >
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to asc.
> > >
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> >
> > **Returns**
> >
> > > *civis.response.PaginatedResponse*
> > >
> > > > - id : integer
> > > > - **public** [boolean] If the template is public or not.
> > > > - **script_id** [integer] The id of the script that this template uses.
> > > > - **user_context** [string] The user context of the script that this template uses.
> > > > - **name** [string] The name of the template.
> > > > - **category** [string] The category of this template.
> > > > - created_at : string/time
> > > > - updated_at : string/time
> > > > - **use_count** [integer] The number of uses of this template.
> > > > - **ui_report_id** [integer] The id of the report that this template uses.
> > > > - **tech_reviewed** [boolean] Whether this template has been audited by Civis for security vulnerability and correctness.
> > > > - **archived** [boolean] Whether the template has been archived.
> > > > - **author** [dict::]
> > > >
> > > > > - **id** [integer] The ID of this user.

> > > – **name** [string] This user's name.
> > >
> > > – **username** [string] This user's username.
> > >
> > > – **initials** [string] This user's initials.
> > >
> > > – **online** [boolean] Whether this user is online.

list_scripts_dependencies(*id*, ***, *user_id='DEFAULT'*)

> List dependent objects for this object
>
> > **Parameters**
> > > **id** [integer] The ID of the resource that is shared.
> > > **user_id** [integer, optional] ID of target user
> >
> > **Returns**
> > > *civis.response.Response*
> > >
> > > > - **object_type** [string] Dependent object type
> > > > - **fco_type** [string] Human readable dependent object type
> > > > - **id** [integer] Dependent object ID
> > > > - **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > > > - **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > > > - **shareable** [boolean] Whether or not the requesting user can share this object.

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

> List the projects a Script Template belongs to
>
> > **Parameters**
> > > **id** [integer] The ID of the Script Template.
> > > **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
> >
> > **Returns**
> > > *civis.response.Response*
> > >
> > > > - **id** [integer] The ID for this project.
> > > > - **author** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > - **name** [string] The name of this project.
> > > > - **description** [string] A description of the project.
> > > > - **users** [list::] Users who can see the project. - id : integer
> > > >
> > > > > The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > - auto_share : boolean
> > > > - created_at : string/time

- updated_at : string/time
- **archived** [string] The archival status of the requested item(s).

list_scripts_shares(*id*)

List users and groups permissioned on this object

> **Parameters**
>
> > **id** [integer] The ID of the resource that is shared.
>
> **Returns**
>
> > [*civis.response.Response*](#)
> >
> > - **readers** [dict::]
> >
> >   - **users** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> >   - **groups** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> > - **writers** [dict::]
> >
> >   - **users** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> >   - **groups** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> > - **owners** [dict::]
> >
> >   - **users** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> >   - **groups** [list::]
> >
> >   * id : integer
> >
> >   * name : string
> >
> > - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch_reports(*id*, *, *name='DEFAULT'*, *category='DEFAULT'*, *archived='DEFAULT'*, *code_body='DEFAULT'*, *provide_api_key='DEFAULT'*)

Update some attributes of this Report Template

> **Parameters**
>
> > **id** [integer]
> >
> > **name** [string, optional] The name of the template.
> >
> > **category** [string, optional] The category of this report template. Can be left blank. Acceptable values are: dataset-viz
> >
> > **archived** [boolean, optional] Whether the template has been archived.
> >
> > **code_body** [string, optional] The code for the Template body.

> > > **provide_api_key** [boolean, optional] Whether reports based on this template request
> > > an API Key from the report viewer.

> > **Returns**
> > > *civis.response.Response*

> > > > • id : integer
> > > > • **name** [string] The name of the template.
> > > > • **category** [string] The category of this report template. Can be left blank.
> > > > > Acceptable values are: dataset-viz
> > > > • created_at : string/time
> > > > • updated_at : string/time
> > > > • **use_count** [integer] The number of uses of this template.
> > > > • **archived** [boolean] Whether the template has been archived.
> > > > • **tech_reviewed** [boolean] Whether this template has been audited by Civis
> > > > > for security vulnerability and correctness.
> > > > • **author** [dict::]

> > > > > – **id** [integer] The ID of this user.

> > > > > – **name** [string] This user's name.

> > > > > – **username** [string] This user's username.

> > > > > – **initials** [string] This user's initials.

> > > > > – **online** [boolean] Whether this user is online.
> > > > • **auth_code_url** [string] A URL to the template's stored code body.
> > > > • **provide_api_key** [boolean] Whether reports based on this template re-
> > > > > quest an API Key from the report viewer.
> > > > • **hidden** [boolean] The hidden status of the item.

patch_scripts(*id*, ***, *name='DEFAULT'*, *note='DEFAULT'*, *ui_report_id='DEFAULT'*,
> *archived='DEFAULT'*)

> Update some attributes of this Script Template

> > **Parameters**
> > > **id** [integer]
> > > **name** [string, optional] The name of the template.
> > > **note** [string, optional] A note describing what this template is used for; custom scripts
> > > > created off this template will display this description.
> > > **ui_report_id** [integer, optional] The id of the report that this template uses.
> > > **archived** [boolean, optional] Whether the template has been archived.

> > **Returns**
> > > *civis.response.Response*

> > > > • id : integer
> > > > • **public** [boolean] If the template is public or not.
> > > > • **script_id** [integer] The id of the script that this template uses.
> > > > • **script_type** [string] The type of the template's backing script (e.g SQL,
> > > > > Container, Python, R, JavaScript)
> > > > • **user_context** [string] The user context of the script that this template uses.
> > > > • **params** [list::] A definition of the parameters that this template's backing
> > > > > script accepts in the arguments field. - name : string

> > > > > > The variable's name as used within your code.

> > > > > – **label** [string] The label to present to users when asking them
> > > > > > for the value.

> > > > > – **description** [string] A short sentence or fragment describing
> > > > > > this parameter to the end user.

- **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom

- **required** [boolean] Whether this param is required.

- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **name** [string] The name of the template.
- **category** [string] The category of this template.
- **note** [string] A note describing what this template is used for; custom scripts created off this template will display this description.
- created_at : string/time
- updated_at : string/time
- **use_count** [integer] The number of uses of this template.
- **ui_report_id** [integer] The id of the report that this template uses.
- **tech_reviewed** [boolean] Whether this template has been audited by Civis for security vulnerability and correctness.
- **archived** [boolean] Whether the template has been archived.
- **hidden** [boolean] The hidden status of the item.
- **author** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

post_reports(*name*, *code_body*, ***, *category='DEFAULT'*, *archived='DEFAULT'*, *provide_api_key='DEFAULT'*, *hidden='DEFAULT'*)

    Create a Report Template

        Parameters

            name [string] The name of the template.

            code_body [string] The code for the Template body.

            category [string, optional] The category of this report template. Can be left blank. Acceptable values are: dataset-viz

            archived [boolean, optional] Whether the template has been archived.

            provide_api_key [boolean, optional] Whether reports based on this template request an API Key from the report viewer.

            hidden [boolean, optional] The hidden status of the item.

        Returns

            civis.response.Response

- id : integer
- **name** [string] The name of the template.
- **category** [string] The category of this report template. Can be left blank.
  Acceptable values are: dataset-viz
- created_at : string/time
- updated_at : string/time
- **use_count** [integer] The number of uses of this template.
- **archived** [boolean] Whether the template has been archived.
- **tech_reviewed** [boolean] Whether this template has been audited by Civis
  for security vulnerability and correctness.
- **author** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.
- **auth_code_url** [string] A URL to the template's stored code body.
- **provide_api_key** [boolean] Whether reports based on this template request an API Key from the report viewer.
- **hidden** [boolean] The hidden status of the item.

post_reports_review(*id*, *status*)

Review a template for security vulnerability and correctness (admin-only)

Parameters

id [integer] The ID of the item.

status [boolean] Whether this item has been reviewed.

Returns

civis.response.Response

- id : integer
- **name** [string] The name of the template.
- **category** [string] The category of this report template. Can be left blank.
  Acceptable values are: dataset-viz
- created_at : string/time
- updated_at : string/time
- **use_count** [integer] The number of uses of this template.
- **archived** [boolean] Whether the template has been archived.
- **tech_reviewed** [boolean] Whether this template has been audited by Civis
  for security vulnerability and correctness.
- **author** [dict::]

  – **id** [integer] The ID of this user.

  – **name** [string] This user's name.

  – **username** [string] This user's username.

  – **initials** [string] This user's initials.

  – **online** [boolean] Whether this user is online.
- **auth_code_url** [string] A URL to the template's stored code body.
- **provide_api_key** [boolean] Whether reports based on this template request an API Key from the report viewer.
- **hidden** [boolean] The hidden status of the item.

post_scripts(*script_id*, *name*, ***, *note='DEFAULT'*, *ui_report_id='DEFAULT'*, *archived='DEFAULT'*, *hidden='DEFAULT'*)

> Create a Script Template
>
> > **Parameters**
> >
> > > **script_id** [integer] The id of the script that this template uses.
> > >
> > > **name** [string] The name of the template.
> > >
> > > **note** [string, optional] A note describing what this template is used for; custom scripts created off this template will display this description.
> > >
> > > **ui_report_id** [integer, optional] The id of the report that this template uses.
> > >
> > > **archived** [boolean, optional] Whether the template has been archived.
> > >
> > > **hidden** [boolean, optional] The hidden status of the item.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - id : integer
> > > > - **public** [boolean] If the template is public or not.
> > > > - **script_id** [integer] The id of the script that this template uses.
> > > > - **script_type** [string] The type of the template's backing script (e.g SQL, Container, Python, R, JavaScript)
> > > > - **user_context** [string] The user context of the script that this template uses.
> > > > - **params** [list::] A definition of the parameters that this template's backing script accepts in the arguments field. - name : string
> > > >
> > > > > The variable's name as used within your code.
> > > > >
> > > > > – **label** [string] The label to present to users when asking them for the value.
> > > > >
> > > > > – **description** [string] A short sentence or fragment describing this parameter to the end user.
> > > > >
> > > > > – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
> > > > >
> > > > > – **required** [boolean] Whether this param is required.
> > > > >
> > > > > – **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.
> > > > >
> > > > > – **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.
> > > > >
> > > > > – **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*
> > > >
> > > > - **name** [string] The name of the template.
> > > > - **category** [string] The category of this template.
> > > > - **note** [string] A note describing what this template is used for; custom scripts created off this template will display this description.
> > > > - created_at : string/time
> > > > - updated_at : string/time
> > > > - **use_count** [integer] The number of uses of this template.

- **ui_report_id** [integer] The id of the report that this template uses.
- **tech_reviewed** [boolean] Whether this template has been audited by Civis
    for security vulnerability and correctness.
- **archived** [boolean] Whether the template has been archived.
- **hidden** [boolean] The hidden status of the item.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

post_scripts_review(*id*, *status*)

> Review a template for security vulnerability and correctness (admin-only)
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the item.
> > > **status** [boolean] Whether this item has been reviewed.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > - id : integer
> > > > - **public** [boolean] If the template is public or not.
> > > > - **script_id** [integer] The id of the script that this template uses.
> > > > - **script_type** [string] The type of the template's backing script (e.g SQL,
> > > >   Container, Python, R, JavaScript)
> > > > - **user_context** [string] The user context of the script that this template uses.
> > > > - **params** [list::] A definition of the parameters that this template's backing
> > > >   script accepts in the arguments field. - name : string
> > > >
> > > > > The variable's name as used within your code.
> > > >
> > > > > - **label** [string] The label to present to users when asking them
> > > > >   for the value.
> > > >
> > > > > - **description** [string] A short sentence or fragment describing
> > > > >   this parameter to the end user.
> > > >
> > > > > - **type** [string] The type of parameter. Valid options:
> > > > >   string, multi_line_string, integer, float, bool, file, table,
> > > > >   database, credential_aws, credential_redshift, or creden-
> > > > >   tial_custom
> > > >
> > > > > - **required** [boolean] Whether this param is required.
> > > >
> > > > > - **value** [string] The value you would like to set this param to.
> > > > >   Setting this value makes this parameter a fixed param.
> > > >
> > > > > - **default** [string] If an argument for this parameter is not de-
> > > > >   fined, 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.
> > > >
> > > > > - **allowed_values** [list] The possible values this parameter can
> > > > >   take, effectively making this an enumerable parameter.

> Allowed values is an array of hashes of the following
> format: *{label: 'Import', 'value': 'import'}*

- **name** [string] The name of the template.
- **category** [string] The category of this template.
- **note** [string] A note describing what this template is used for; custom
  scripts created off this template will display this description.
- created_at : string/time
- updated_at : string/time
- **use_count** [integer] The number of uses of this template.
- **ui_report_id** [integer] The id of the report that this template uses.
- **tech_reviewed** [boolean] Whether this template has been audited by Civis
  for security vulnerability and correctness.
- **archived** [boolean] Whether the template has been archived.
- **hidden** [boolean] The hidden status of the item.
- **author** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.

put_reports(*id*, *name*, *code_body*, *, *category='DEFAULT'*, *archived='DEFAULT'*,
  provide_api_key='DEFAULT')

  Replace all attributes of this Report Template

  Parameters

  id [integer]

  name [string] The name of the template.

  code_body [string] The code for the Template body.

  category [string, optional] The category of this report template. Can be left blank.
  Acceptable values are: dataset-viz

  archived [boolean, optional] Whether the template has been archived.

  provide_api_key [boolean, optional] Whether reports based on this template request
  an API Key from the report viewer.

  Returns

  civis.response.Response

  - id : integer
  - **name** [string] The name of the template.
  - **category** [string] The category of this report template. Can be left blank.
    Acceptable values are: dataset-viz
  - created_at : string/time
  - updated_at : string/time
  - **use_count** [integer] The number of uses of this template.
  - **archived** [boolean] Whether the template has been archived.
  - **tech_reviewed** [boolean] Whether this template has been audited by Civis
    for security vulnerability and correctness.
  - **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

> > > > – **initials** [string] This user's initials.
> > > >
> > > > – **online** [boolean] Whether this user is online.
> > >
> > > • **auth_code_url** [string] A URL to the template's stored code body.
> > > • **provide_api_key** [boolean] Whether reports based on this template request an API Key from the report viewer.
> > > • **hidden** [boolean] The hidden status of the item.

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

> Set the permissions groups has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > > **group_ids** [list] An array of one or more group IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **readers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > >
> > > > • **writers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > >
> > > > • **owners** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > * id : integer
> > > > > >
> > > > > > * name : string
> > > >
> > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

>   Set the permissions users have on this object

>   > **Parameters**

>   > > **id** [integer] The ID of the resource that is shared.
>   > > **user_ids** [list] An array of one or more user IDs.
>   > > **permission_level** [string] Options are: "read", "write", or "manage".
>   > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
>   > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.

>   > **Returns**

>   > [*civis.response.Response*](#)

>   > > > • **readers** [dict::]

>   > > > > – **users** [list::]

>   > > > > > ∗ id : integer

>   > > > > > ∗ name : string

>   > > > > – **groups** [list::]

>   > > > > > ∗ id : integer

>   > > > > > ∗ name : string

>   > > > • **writers** [dict::]

>   > > > > – **users** [list::]

>   > > > > > ∗ id : integer

>   > > > > > ∗ name : string

>   > > > > – **groups** [list::]

>   > > > > > ∗ id : integer

>   > > > > > ∗ name : string

>   > > > • **owners** [dict::]

>   > > > > – **users** [list::]

>   > > > > > ∗ id : integer

>   > > > > > ∗ name : string

>   > > > > – **groups** [list::]

>   > > > > > ∗ id : integer

>   > > > > > ∗ name : string

>   > > > • **total_user_shares** [integer] For owners, the number of total users shared.
>   > > > For writers and readers, the number of visible users shared.
>   > > > • **total_group_shares** [integer] For owners, the number of total groups
>   > > > shared. For writers and readers, the number of visible groups shared.

put_reports_transfer(*id*, *user_id*, *include_dependencies*, *, *email_body='DEFAULT'*,
                                    send_email='DEFAULT')

>   Transfer ownership of this object to another user

>   > **Parameters**

>   > > **id** [integer] The ID of the resource that is shared.
>   > > **user_id** [integer] ID of target user
>   > > **include_dependencies** [boolean] Whether or not to give manage permissions on all
>   > > dependencies

> > **email_body** [string, optional] Custom body text for e-mail sent on transfer.
> > **send_email** [boolean, optional] Send email to the target user of the transfer?
>
> > **Returns**
> > *civis.response.Response*
>
> > > • **dependencies** [list::] Dependent objects for this object - object_type : string
> > >
> > > > Dependent object type
> > >
> > > > – **fco_type** [string] Human readable dependent object type
> > > >
> > > > – **id** [integer] Dependent object ID
> > > >
> > > > – **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> > > >
> > > > – **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> > > >
> > > > – **shared** [boolean] Whether dependent object was successfully shared with target user

put_scripts(*id*, *name*, ***, *note='DEFAULT'*, *ui_report_id='DEFAULT'*, *archived='DEFAULT'*)

> Replace all attributes of this Script Template
>
> > **Parameters**
> > **id** [integer]
> > **name** [string] The name of the template.
> > **note** [string, optional] A note describing what this template is used for; custom scripts created off this template will display this description.
> > **ui_report_id** [integer, optional] The id of the report that this template uses.
> > **archived** [boolean, optional] Whether the template has been archived.
> >
> > **Returns**
> > *civis.response.Response*
>
> > > • id : integer
> > > • **public** [boolean] If the template is public or not.
> > > • **script_id** [integer] The id of the script that this template uses.
> > > • **script_type** [string] The type of the template's backing script (e.g SQL, Container, Python, R, JavaScript)
> > > • **user_context** [string] The user context of the script that this template uses.
> > > • **params** [list::] A definition of the parameters that this template's backing script accepts in the arguments field. - name : string
> > >
> > > > The variable's name as used within your code.
> > >
> > > > – **label** [string] The label to present to users when asking them for the value.
> > > >
> > > > – **description** [string] A short sentence or fragment describing this parameter to the end user.
> > > >
> > > > – **type** [string] The type of parameter. Valid options: string, multi_line_string, integer, float, bool, file, table, database, credential_aws, credential_redshift, or credential_custom
> > > >
> > > > – **required** [boolean] Whether this param is required.

- **value** [string] The value you would like to set this param to. Setting this value makes this parameter a fixed param.

- **default** [string] If an argument for this parameter is not defined, it will use this default value. Use true, True, t, y, yes, or 1 for true bool's or false, False, f, n, no, or 0 for false bool's. Cannot be used for parameters that are required or a credential type.

- **allowed_values** [list] The possible values this parameter can take, effectively making this an enumerable parameter. Allowed values is an array of hashes of the following format: *{label: 'Import', 'value': 'import'}*

- **name** [string] The name of the template.
- **category** [string] The category of this template.
- **note** [string] A note describing what this template is used for; custom scripts created off this template will display this description.
- created_at : string/time
- updated_at : string/time
- **use_count** [integer] The number of uses of this template.
- **ui_report_id** [integer] The id of the report that this template uses.
- **tech_reviewed** [boolean] Whether this template has been audited by Civis for security vulnerability and correctness.
- **archived** [boolean] Whether the template has been archived.
- **hidden** [boolean] The hidden status of the item.
- **author** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

put_scripts_projects(*id*, *project_id*)

    Add a Script Template to a project

        Parameters

            id [integer] The ID of the Script Template.

            project_id [integer] The ID of the project.

        Returns

            None Response code 204: success

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

    Set the permissions groups has on this object

        Parameters

            id [integer] The ID of the resource that is shared.

            group_ids [list] An array of one or more group IDs.

            permission_level [string] Options are: "read", "write", or "manage".

            share_email_body [string, optional] Custom body text for e-mail sent on a share.

            send_shared_email [boolean, optional] Send email to the recipients of a share.

        Returns

            civis.response.Response

                - **readers** [dict::]

> > > > > > > – **users** [list::]

> > > > > > > > > ∗ id : integer

> > > > > > > > > ∗ name : string

> > > > > > > – **groups** [list::]

> > > > > > > > > ∗ id : integer

> > > > > > > > > ∗ name : string

> > > > > > • **writers** [dict::]

> > > > > > > – **users** [list::]

> > > > > > > > > ∗ id : integer

> > > > > > > > > ∗ name : string

> > > > > > > – **groups** [list::]

> > > > > > > > > ∗ id : integer

> > > > > > > > > ∗ name : string

> > > > > > • **owners** [dict::]

> > > > > > > – **users** [list::]

> > > > > > > > > ∗ id : integer

> > > > > > > > > ∗ name : string

> > > > > > > – **groups** [list::]

> > > > > > > > > ∗ id : integer

> > > > > > > > > ∗ name : string

> > > > > > • **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.

> > > > > > • **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

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

> Set the permissions users have on this object

> > **Parameters**

> > > **id** [integer] The ID of the resource that is shared.

> > > **user_ids** [list] An array of one or more user IDs.

> > > **permission_level** [string] Options are: "read", "write", or "manage".

> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.

> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.

> > **Returns**

> > [*civis.response.Response*](#)

> > > > • **readers** [dict::]

> > > > > – **users** [list::]

> > > > > > > ∗ id : integer

> > > > > > > ∗ name : string

> > > > > > – **groups** [list::]

> > > > > > > ∗ id : integer

> > > > > > > ∗ name : string

- **writers** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **owners** [dict::]

    - **users** [list::]

        * id : integer

        * name : string

    - **groups** [list::]

        * id : integer

        * name : string

- **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
- **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

put_scripts_transfer(*id*, *user_id*, *include_dependencies*, *, *email_body='DEFAULT'*, *send_email='DEFAULT'*)

Transfer ownership of this object to another user

Parameters

id [integer] The ID of the resource that is shared.

user_id [integer] ID of target user

include_dependencies [boolean] Whether or not to give manage permissions on all dependencies

email_body [string, optional] Custom body text for e-mail sent on transfer.

send_email [boolean, optional] Send email to the target user of the transfer?

Returns

[*civis.response.Response*](#)

- **dependencies** [list::] Dependent objects for this object - object_type : string

    Dependent object type

    - **fco_type** [string] Human readable dependent object type

    - **id** [integer] Dependent object ID

    - **name** [string] Dependent object name, or nil if the requesting user cannot read this object

    - **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user

    - **shared** [boolean] Whether dependent object was successfully shared with target user

Users

class Users(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.users.list(...)
```

Methods

delete_api_keys(id, key_id)	Revoke the specified API key
delete_me_favorites(id)	Unfavorite an item
delete_me_superadmin()	Disables Superadmin Mode for the current user
delete_sessions(id)	Terminate all of the user's active sessions (must be an admin or client user admin)
get(id)	Show info about a user
get_api_keys(id, key_id)	Show the specified API key
list(*[, feature_flag, account_status, ...])	List users
list_api_keys(id, *[, limit, page_num, ...])	Show API keys belonging to the specified user
list_me()	Show info about the logged-in user
list_me_favorites(*[, object_id, ...])	List Favorites
list_me_ui()	UI configuration for logged-in user
patch(id, *[, name, email, active, ...])	Update info about a user (must be an admin or client user admin)
patch_me(*[, preferences, ...])	Update info about the logged-in user
post(name, email, primary_group_id, user, *)	Create a new user (must be an admin or client user admin)
post_api_keys(id, expires_in, name, *[, ...])	Create a new API key belonging to the logged-in user
post_me_favorites(object_id, object_type)	Favorite an item
post_me_superadmin()	Enables Superadmin Mode for the current user
post_unsuspend(id)	Unsuspends 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**
> > *civis.response.Response*
> > > • **id** [integer] The ID of the API key.
> > > • **name** [string] The name of the API key.
> > > • **expires_at** [string/date-time] The date and time when the key expired.
> > > • **created_at** [string/date-time] The date and time when the key was created.
> > > • **revoked_at** [string/date-time] The date and time when the key was revoked.

- **last_used_at** [string/date-time] The date and time when the key was last used.
- **scopes** [list] The scopes which the key is permissioned on.
- **use_count** [integer] The number of times the key has been used.
- **expired** [boolean] True if the key has expired.
- **active** [boolean] True if the key has neither expired nor been revoked.
- **constraints** [list::] Constraints on the abilities of the created key - constraint : string

  The path matcher of the constraint.

  – **constraint_type** [string] The type of constraint (exact/prefix/regex/verb).

  – **get_allowed** [boolean] Whether the constraint allows GET requests.

  – **head_allowed** [boolean] Whether the constraint allows HEAD requests.

  – **post_allowed** [boolean] Whether the constraint allows POST requests.

  – **put_allowed** [boolean] Whether the constraint allows PUT requests.

  – **patch_allowed** [boolean] Whether the constraint allows PATCH requests.

  – **delete_allowed** [boolean] Whether the constraint allows DELETE requests.

delete_me_favorites(*id*)

    Unfavorite an item

        Parameters

            id [integer] The id of the favorite.

        Returns

            None Response code 204: success

delete_me_superadmin()

    Disables Superadmin Mode for the current user

        Returns

            civis.response.Response

- **id** [integer] The ID of this user.
- **name** [string] This user's name.
- **email** [string] This user's email address.
- **username** [string] This user's username.
- **initials** [string] This user's initials.
- **last_checked_announcements** [string/date-time] The date and time at which the user last checked their announcements.
- **feature_flags** [dict] The feature flag settings for this user.
- **roles** [list] The roles this user has, listed by slug.
- **preferences** [dict] This user's preferences.
- **custom_branding** [string] The branding of Platform for this user.
- **primary_group_id** [integer] The ID of the primary group of this user.
- **groups** [list::] An array of all the groups this user is in. - id : integer

  The ID of this group.

> > > > – **name** [string] The name of this group.
> > > >
> > > > – **organization_id** [integer] The ID of the organization associated with this group.
> > > >
> > > > – **organization_name** [string] The name of the organization associated with this group.
> > >
> > > - **organization_name** [string] The name of the organization the user belongs to.
> > > - **organization_slug** [string] The slug of the organization the user belongs to.
> > > - **organization_default_theme_id** [integer] The ID of the organizations's default theme.
> > > - **created_at** [string/date-time] The date and time when the user was created.
> > > - **sign_in_count** [integer] The number of times the user has signed in.
> > > - **assuming_role** [boolean] Whether the user is assuming a role or not.
> > > - **assuming_admin** [boolean] Whether the user is assuming admin.
> > > - **assuming_admin_expiration** [string/date-time] When the user's admin role is set to expire.
> > > - **superadmin_mode_expiration** [string/date-time] The user is in superadmin mode when set to a DateTime. The user is not in superadmin mode when set to null.
> > > - **disable_non_compliant_fedramp_features** [boolean] Whether to disable non-compliant fedramp features.
> > > - **created_by_id** [integer] The ID of the user who created this user.
> > > - **last_updated_by_id** [integer] The ID of the user who last updated this user.

delete_sessions(*id*)

Terminate all of the user's active sessions (must be an admin or client user admin)

> **Parameters**
>
> > **id** [integer] The ID of this user.
>
> **Returns**
>
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID of this user.
> > > - **user** [string] The username of this user.
> > > - **name** [string] The name of this user.
> > > - **email** [string] The email of this user.
> > > - **active** [boolean] The account status of this user.
> > > - **primary_group_id** [integer] The ID of the primary group of this user.
> > > - **groups** [list::] An array of all the groups this user is in. - id : integer
> > >
> > > > The ID of this group.
> > > >
> > > > – **name** [string] The name of this group.
> > > >
> > > > – **organization_id** [integer] The ID of the organization associated with this group.
> > > >
> > > > – **organization_name** [string] The name of the organization associated with this group.
> > >
> > > - **city** [string] The city of this user.
> > > - **state** [string] The state of this user.
> > > - **time_zone** [string] The time zone of this user.
> > > - **initials** [string] The initials of this user.

- **department** [string] The department of this user.
- **title** [string] The title of this user.
- **github_username** [string] The GitHub username of this user.
- **prefers_sms_otp** [boolean] The preference for phone authorization of this user
- **vpn_enabled** [boolean] The availability of vpn for this user.
- **sso_disabled** [boolean] The availability of SSO for this user.
- **otp_required_for_login** [boolean] The two factor authentication requirement for this user.
- **exempt_from_org_sms_otp_disabled** [boolean] Whether the user has SMS OTP enabled on an individual level. This field does not matter if the org does not have SMS OTP disabled.
- **sms_otp_allowed** [boolean] Whether the user is allowed to receive two factor authentication codes via SMS.
- **robot** [boolean] Whether the user is a robot.
- **phone** [string] The phone number of this user.
- **organization_slug** [string] The slug of the organization the user belongs to.
- **organization_sso_disable_capable** [boolean] The user's organization's ability to disable sso for their users.
- **organization_login_type** [string] The user's organization's login type.
- **organization_sms_otp_disabled** [boolean] Whether the user's organization has SMS OTP disabled.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **created_at** [string/date-time] The date and time when the user was created.
- **updated_at** [string/date-time] The date and time when the user was last updated.
- **last_seen_at** [string/date-time] The date and time when the user last visited Platform.
- **suspended** [boolean] Whether the user is suspended due to inactivity.
- **created_by_id** [integer] The ID of the user who created this user.
- **last_updated_by_id** [integer] The ID of the user who last updated this user.

get(*id*)

> Show info about a user
>
> > **Parameters**
> >
> > > **id** [integer] The ID of this user.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > - **id** [integer] The ID of this user.
> > > - **user** [string] The username of this user.
> > > - **name** [string] The name of this user.
> > > - **email** [string] The email of this user.
> > > - **active** [boolean] The account status of this user.
> > > - **primary_group_id** [integer] The ID of the primary group of this user.
> > > - **groups** [list::] An array of all the groups this user is in. - id : integer
> > >
> > > > The ID of this group.
> > >
> > > > – **name** [string] The name of this group.
> > >
> > > > – **organization_id** [integer] The ID of the organization associ-

> > > ated with this group.

> > > > – **organization_name** [string] The name of the organization associated with this group.

- **city** [string] The city of this user.
- **state** [string] The state of this user.
- **time_zone** [string] The time zone of this user.
- **initials** [string] The initials of this user.
- **department** [string] The department of this user.
- **title** [string] The title of this user.
- **github_username** [string] The GitHub username of this user.
- **prefers_sms_otp** [boolean] The preference for phone authorization of this user
- **vpn_enabled** [boolean] The availability of vpn for this user.
- **sso_disabled** [boolean] The availability of SSO for this user.
- **otp_required_for_login** [boolean] The two factor authentication requirement for this user.
- **exempt_from_org_sms_otp_disabled** [boolean] Whether the user has SMS OTP enabled on an individual level. This field does not matter if the org does not have SMS OTP disabled.
- **sms_otp_allowed** [boolean] Whether the user is allowed to receive two factor authentication codes via SMS.
- **robot** [boolean] Whether the user is a robot.
- **phone** [string] The phone number of this user.
- **organization_slug** [string] The slug of the organization the user belongs to.
- **organization_sso_disable_capable** [boolean] The user's organization's ability to disable sso for their users.
- **organization_login_type** [string] The user's organization's login type.
- **organization_sms_otp_disabled** [boolean] Whether the user's organization has SMS OTP disabled.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **created_at** [string/date-time] The date and time when the user was created.
- **updated_at** [string/date-time] The date and time when the user was last updated.
- **last_seen_at** [string/date-time] The date and time when the user last visited Platform.
- **suspended** [boolean] Whether the user is suspended due to inactivity.
- **created_by_id** [integer] The ID of the user who created this user.
- **last_updated_by_id** [integer] The ID of the user who last updated 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**

> > *civis.response.Response*

> > > - **id** [integer] The ID of the API key.
> > > - **name** [string] The name of the API key.
> > > - **expires_at** [string/date-time] The date and time when the key expired.

- **created_at** [string/date-time] The date and time when the key was created.
- **revoked_at** [string/date-time] The date and time when the key was revoked.
- **last_used_at** [string/date-time] The date and time when the key was last used.
- **scopes** [list] The scopes which the key is permissioned on.
- **use_count** [integer] The number of times the key has been used.
- **expired** [boolean] True if the key has expired.
- **active** [boolean] True if the key has neither expired nor been revoked.
- **constraints** [list::] Constraints on the abilities of the created key - constraint : string

   The path matcher of the constraint.

   – **constraint_type** [string] The type of constraint (exact/prefix/regex/verb).

   – **get_allowed** [boolean] Whether the constraint allows GET requests.

   – **head_allowed** [boolean] Whether the constraint allows HEAD requests.

   – **post_allowed** [boolean] Whether the constraint allows POST requests.

   – **put_allowed** [boolean] Whether the constraint allows PUT requests.

   – **patch_allowed** [boolean] Whether the constraint allows PATCH requests.

   – **delete_allowed** [boolean] Whether the constraint allows DELETE requests.

list(*, *feature_flag='DEFAULT'*, *account_status='DEFAULT'*, *query='DEFAULT'*, *group_id='DEFAULT'*, *group_ids='DEFAULT'*, *organization_id='DEFAULT'*, *exclude_groups='DEFAULT'*, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)

List users

> **Parameters**
>> **feature_flag** [string, optional] Return users that have a feature flag enabled.
>>
>> **account_status** [string, optional] The account status by which to filter users. May be one of "active", "inactive", or "all". Defaults to active.
>>
>> **query** [string, optional] Return users who match the given query, based on name, user, email, and id.
>>
>> **group_id** [integer, optional] The ID of the group by which to filter users. Cannot be present if group_ids is.
>>
>> **group_ids** [array, optional] The IDs of the groups by which to filter users. Cannot be present if group_id is.
>>
>> **organization_id** [integer, optional] The ID of the organization by which to filter users.
>>
>> **exclude_groups** [boolean, optional] Whether or to exclude users' groups. Default: false.
>>
>> **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 10000.
>>
>> **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
>>
>> **order** [string, optional] The field on which to order the result set. Defaults to name. Must be one of: name, user.

> **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to asc.
>
> **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.

Returns

civis.response.PaginatedResponse

- **id** [integer] The ID of this user.
- **user** [string] The username of this user.
- **name** [string] The name of this user.
- **email** [string] The email of this user.
- **active** [boolean] The account status of this user.
- **primary_group_id** [integer] The ID of the primary group of this user.
- **groups** [list::] An array of all the groups this user is in. - id : integer

    The ID of this group.

  - **name** [string] The name of this group.
  - **organization_id** [integer] The ID of the organization associated with this group.
  - **organization_name** [string] The name of the organization associated with this group.

- **created_at** [string/date-time] The date and time when the user was created.
- **current_sign_in_at** [string/date-time] The date and time when the user's current session began.
- **updated_at** [string/date-time] The date and time when the user was last updated.
- **last_seen_at** [string/date-time] The date and time when the user last visited Platform.
- **suspended** [boolean] Whether the user is suspended due to inactivity.
- **created_by_id** [integer] The ID of the user who created this user.
- **last_updated_by_id** [integer] The ID of the user who last updated this user.

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

Show API keys belonging to the specified user

Parameters

id [string] The ID of the user or 'me'.

limit [integer, optional] Number of results to return. Defaults to its maximum of 50.

page_num [integer, optional] Page number of the results to return. Defaults to the first page, 1.

order [string, optional] The field on which to order the result set. Defaults to id. Must be one of: id.

order_dir [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.

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

Returns

civis.response.PaginatedResponse

- **id** [integer] The ID of the API key.

- **name** [string] The name of the API key.
- **expires_at** [string/date-time] The date and time when the key expired.
- **created_at** [string/date-time] The date and time when the key was created.
- **revoked_at** [string/date-time] The date and time when the key was revoked.
- **last_used_at** [string/date-time] The date and time when the key was last used.
- **scopes** [list] The scopes which the key is permissioned on.
- **use_count** [integer] The number of times the key has been used.
- **expired** [boolean] True if the key has expired.
- **active** [boolean] True if the key has neither expired nor been revoked.
- **constraint_count** [integer] The number of constraints on the created key

`list_me()`

Show info about the logged-in user

> **Returns**
>
> > *civis.response.Response*
> >
> > - **id** [integer] The ID of this user.
> > - **name** [string] This user's name.
> > - **email** [string] This user's email address.
> > - **username** [string] This user's username.
> > - **initials** [string] This user's initials.
> > - **last_checked_announcements** [string/date-time] The date and time at which the user last checked their announcements.
> > - **feature_flags** [dict] The feature flag settings for this user.
> > - **roles** [list] The roles this user has, listed by slug.
> > - **preferences** [dict] This user's preferences.
> > - **custom_branding** [string] The branding of Platform for this user.
> > - **primary_group_id** [integer] The ID of the primary group of this user.
> > - **groups** [list::] An array of all the groups this user is in. - id : integer
> >
> > > The ID of this group.
> >
> > > - **name** [string] The name of this group.
> > >
> > > - **organization_id** [integer] The ID of the organization associated with this group.
> > >
> > > - **organization_name** [string] The name of the organization associated with this group.
> >
> > - **organization_name** [string] The name of the organization the user belongs to.
> > - **organization_slug** [string] The slug of the organization the user belongs to.
> > - **organization_default_theme_id** [integer] The ID of the organizations's default theme.
> > - **created_at** [string/date-time] The date and time when the user was created.
> > - **sign_in_count** [integer] The number of times the user has signed in.
> > - **assuming_role** [boolean] Whether the user is assuming a role or not.
> > - **assuming_admin** [boolean] Whether the user is assuming admin.
> > - **assuming_admin_expiration** [string/date-time] When the user's admin role is set to expire.
> > - **superadmin_mode_expiration** [string/date-time] The user is in superadmin mode when set to a DateTime. The user is not in superadmin

mode when set to null.

- **disable_non_compliant_fedramp_features** [boolean] Whether to disable non-compliant fedramp features.
- **created_by_id** [integer] The ID of the user who created this user.
- **last_updated_by_id** [integer] The ID of the user who last updated this user.

list_me_favorites(*, *object_id='DEFAULT'*, *object_type='DEFAULT'*, *limit='DEFAULT'*, *page_num='DEFAULT'*, *order='DEFAULT'*, *order_dir='DEFAULT'*, *iterator='DEFAULT'*)

List Favorites

> **Parameters**
>> **object_id** [integer, optional] The id of the object. If specified as a query parameter, must also specify object_type parameter.
>> **object_type** [string, optional] The type of the object that is favorited. Valid options: Project
>> **limit** [integer, optional] Number of results to return. Defaults to 50. Maximum allowed is 1000.
>> **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
>> **order** [string, optional] The field on which to order the result set. Defaults to created_at. Must be one of: created_at, object_type, object_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**
> *civis.response.PaginatedResponse*
>> - **id** [integer] The id of the favorite.
>> - **object_id** [integer] The id of the object. If specified as a query parameter, must also specify object_type parameter.
>> - **object_type** [string] The type of the object that is favorited. Valid options: Project
>> - **object_name** [string] The name of the object that is favorited.
>> - **created_at** [string/time] The time this favorite was created.

list_me_ui()

UI configuration for logged-in user

> **Returns**
> *civis.response.Response*
>> - **id** [integer] The ID of this user.
>> - **navigation_menus** [dict] Navigation menus visible to this user.
>> - **user_menus** [dict] User profile menu items available to this user.
>> - **user_type** [dict::]
>>
>>> - **vendor** [boolean] This attribute is deprecated
>>>
>>> - **media** [boolean] True if user has access to the Media Optimizer job type.
>>>
>>> - **main_app** [string] This attribute is deprecated
>>>
>>> - **app_count** [integer] This attribute is deprecated
>>>
>>> - **reports_only** [boolean] True if user is a reports-only user.

> > > > > – **reports_creator** [boolean] True if this user is allowed to create HTML reports.
> > > > • **zendesk_token** [string] JSON web token for this user's Zendesk widget.

> > patch(*id*, ***, *name='DEFAULT'*, *email='DEFAULT'*, *active='DEFAULT'*, *primary_group_id='DEFAULT'*, *city='DEFAULT'*, *state='DEFAULT'*, *time_zone='DEFAULT'*, *initials='DEFAULT'*, *department='DEFAULT'*, *title='DEFAULT'*, *prefers_sms_otp='DEFAULT'*, *group_ids='DEFAULT'*, *vpn_enabled='DEFAULT'*, *sso_disabled='DEFAULT'*, *otp_required_for_login='DEFAULT'*, *exempt_from_org_sms_otp_disabled='DEFAULT'*, *robot='DEFAULT'*, *phone='DEFAULT'*, *password='DEFAULT'*)

> > > Update info about a user (must be an admin or client user admin)

> > > > **Parameters**

> > > > > **id** [integer] The ID of this user.
> > > > > **name** [string, optional] The name of this user.
> > > > > **email** [string, optional] The email of this user.
> > > > > **active** [boolean, optional] The account status of this user.
> > > > > **primary_group_id** [integer, optional] The ID of the primary group of this user.
> > > > > **city** [string, optional] The city of this user.
> > > > > **state** [string, optional] The state of this user.
> > > > > **time_zone** [string, optional] The time zone of this user.
> > > > > **initials** [string, optional] The initials of this user.
> > > > > **department** [string, optional] The department of this user.
> > > > > **title** [string, optional] The title of this user.
> > > > > **prefers_sms_otp** [boolean, optional] The preference for phone authorization of this user
> > > > > **group_ids** [list, optional] An array of ids of all the groups this user is in.
> > > > > **vpn_enabled** [boolean, optional] The availability of vpn for this user.
> > > > > **sso_disabled** [boolean, optional] The availability of SSO for this user.
> > > > > **otp_required_for_login** [boolean, optional] The two factor authentication requirement for this user.
> > > > > **exempt_from_org_sms_otp_disabled** [boolean, optional] Whether the user has SMS OTP enabled on an individual level. This field does not matter if the org does not have SMS OTP disabled.
> > > > > **robot** [boolean, optional] Whether the user is a robot.
> > > > > **phone** [string, optional] The phone number of this user.
> > > > > **password** [string, optional] The password of this user.

> > > > **Returns**

> > > > > [*civis.response.Response*](#)

> > > > > > • **id** [integer] The ID of this user.
> > > > > > • **user** [string] The username of this user.
> > > > > > • **name** [string] The name of this user.
> > > > > > • **email** [string] The email of this user.
> > > > > > • **active** [boolean] The account status of this user.
> > > > > > • **primary_group_id** [integer] The ID of the primary group of this user.
> > > > > > • **groups** [list::] An array of all the groups this user is in. - id : integer

> > > > > > > The ID of this group.

> > > > > > > – **name** [string] The name of this group.

> > > > > > > – **organization_id** [integer] The ID of the organization associated with this group.

> > > > > > > – **organization_name** [string] The name of the organization associated with this group.

- **city** [string] The city of this user.
- **state** [string] The state of this user.
- **time_zone** [string] The time zone of this user.
- **initials** [string] The initials of this user.
- **department** [string] The department of this user.
- **title** [string] The title of this user.
- **github_username** [string] The GitHub username of this user.
- **prefers_sms_otp** [boolean] The preference for phone authorization of this user
- **vpn_enabled** [boolean] The availability of vpn for this user.
- **sso_disabled** [boolean] The availability of SSO for this user.
- **otp_required_for_login** [boolean] The two factor authentication requirement for this user.
- **exempt_from_org_sms_otp_disabled** [boolean] Whether the user has SMS OTP enabled on an individual level. This field does not matter if the org does not have SMS OTP disabled.
- **sms_otp_allowed** [boolean] Whether the user is allowed to receive two factor authentication codes via SMS.
- **robot** [boolean] Whether the user is a robot.
- **phone** [string] The phone number of this user.
- **organization_slug** [string] The slug of the organization the user belongs to.
- **organization_sso_disable_capable** [boolean] The user's organization's ability to disable sso for their users.
- **organization_login_type** [string] The user's organization's login type.
- **organization_sms_otp_disabled** [boolean] Whether the user's organization has SMS OTP disabled.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- **created_at** [string/date-time] The date and time when the user was created.
- **updated_at** [string/date-time] The date and time when the user was last updated.
- **last_seen_at** [string/date-time] The date and time when the user last visited Platform.
- **suspended** [boolean] Whether the user is suspended due to inactivity.
- **created_by_id** [integer] The ID of the user who created this user.
- **last_updated_by_id** [integer] The ID of the user who last updated this user.

patch_me(*, *preferences='DEFAULT'*, *last_checked_announcements='DEFAULT'*)
> Update info about the logged-in user
>> **Parameters**
>>> **preferences** [dict, optional::]
>>>> - **app_index_order_field** [string] This attribute is deprecated
>>>> - **app_index_order_dir** [string] This attribute is deprecated
>>>> - **result_index_order_field** [string] Order field for the results index page.
>>>> - **result_index_order_dir** [string] Order direction for the results index page.
>>>> - **result_index_type_filter** [string] Type filter for the results index page.
>>>> - **result_index_author_filter** [string] Author filter for the results index page.
>>>> - **result_index_archived_filter** [string] Archived filter for the results index page.

- **import_index_order_field** [string] Order field for the imports index page.
- **import_index_order_dir** [string] Order direction for the imports index page.
- **import_index_type_filter** [string] Type filter for the imports index page.
- **import_index_author_filter** [string] Author filter for the imports index page.
- **import_index_dest_filter** [string] Destination filter for the imports index page.
- **import_index_status_filter** [string] Status filter for the imports index page.
- **import_index_archived_filter** [string] Archived filter for the imports index page.
- **export_index_order_field** [string] Order field for the exports index page.
- **export_index_order_dir** [string] Order direction for the exports index page.
- **export_index_type_filter** [string] Type filter for the exports index page.
- **export_index_author_filter** [string] Author filter for the exports index page.
- **export_index_status_filter** [string] Status filter for the exports index page.
- **model_index_order_field** [string] Order field for the models index page.
- **model_index_order_dir** [string] Order direction for the models index page.
- **model_index_author_filter** [string] Author filter for the models index page.
- **model_index_status_filter** [string] Status filter for the models index page.
- **model_index_archived_filter** [string] Archived filter for the models index page.
- **model_index_thumbnail_view** [string] Thumbnail view for the models index page.
- **script_index_order_field** [string] Order field for the scripts index page.
- **script_index_order_dir** [string] Order direction for the scripts index page.
- **script_index_type_filter** [string] Type filter for the scripts index page.
- **script_index_author_filter** [string] Author filter for the scripts index page.
- **script_index_status_filter** [string] Status filter for the scripts index page.
- **script_index_archived_filter** [string] Archived filter for the scripts index page.
- **project_index_order_field** [string] Order field for the projects index page.
- **project_index_order_dir** [string] Order direction for the projects index page.
- **project_index_author_filter** [string] Author filter for the projects index page.
- **project_index_archived_filter** [string] Archived filter for the projects index page.
- **report_index_thumbnail_view** [string] Thumbnail view for the reports index page.
- **project_detail_order_field** [string] Order field for projects detail pages.
- **project_detail_order_dir** [string] Order direction for projects detail pages.
- **project_detail_author_filter** [string] Author filter for projects detail pages.

- **project_detail_type_filter** [string] Type filter for projects detail pages.
- **project_detail_archived_filter** [string] Archived filter for the projects detail pages.
- **enhancement_index_order_field** [string] Order field for the enhancements index page.
- **enhancement_index_order_dir** [string] Order direction for the enhancements index page.
- **enhancement_index_author_filter** [string] Author filter for the enhancements index page.
- **enhancement_index_archived_filter** [string] Archived filter for the enhancements index page.
- **preferred_server_id** [integer] ID of preferred server.
- **civis_explore_skip_intro** [boolean] Whether the user is shown steps for each exploration.
- **registration_index_order_field** [string] Order field for the registrations index page.
- **registration_index_order_dir** [string] Order direction for the registrations index page.
- **registration_index_status_filter** [string] Status filter for the registrations index page.
- **upgrade_requested** [string] Whether a free trial upgrade has been requested.
- **welcome_order_field** [string] Order direction for the welcome page.
- **welcome_order_dir** [string] Order direction for the welcome page.
- **welcome_author_filter** [string] Status filter for the welcome page.
- **welcome_status_filter** [string] Status filter for the welcome page.
- **welcome_archived_filter** [string] Status filter for the welcome page.
- **data_pane_width** [string] Width of the data pane when expanded.
- **data_pane_collapsed** [string] Whether the data pane is collapsed.
- **notebook_order_field** [string] Order field for the notebooks page.
- **notebook_order_dir** [string] Order direction for the notebooks page.
- **notebook_author_filter** [string] Author filter for the notebooks page.
- **notebook_archived_filter** [string] Archived filter for the notebooks page.
- **notebook_status_filter** [string] Status filter for the notebooks page.
- **workflow_index_order_field** [string] Order field for the workflows page.
- **workflow_index_order_dir** [string] Order direction for the workflows page.
- **workflow_index_author_filter** [string] Author filter for the workflows page.
- **workflow_index_archived_filter** [string] Archived filter for the workflows page.
- **service_order_field** [string] Order field for the services page.
- **service_order_dir** [string] Order direction for the services page.
- **service_author_filter** [string] Author filter for the services page.
- **service_archived_filter** [string] Archived filter for the services page.
- **assume_role_history** [string] JSON string of previously assumed roles.

last_checked_announcements [string/date-time, optional] The date and time at which the user last checked their announcements.

Returns

civis.response.Response

- **id** [integer] The ID of this user.
- **name** [string] This user's name.
- **email** [string] This user's email address.
- **username** [string] This user's username.

- **initials** [string] This user's initials.
- **last_checked_announcements** [string/date-time] The date and time at which the user last checked their announcements.
- **feature_flags** [dict] The feature flag settings for this user.
- **roles** [list] The roles this user has, listed by slug.
- **preferences** [dict] This user's preferences.
- **custom_branding** [string] The branding of Platform for this user.
- **primary_group_id** [integer] The ID of the primary group of this user.
- **groups** [list::] An array of all the groups this user is in. - id : integer

     The ID of this group.

     – **name** [string] The name of this group.

     – **organization_id** [integer] The ID of the organization associated with this group.

     – **organization_name** [string] The name of the organization associated with this group.

- **organization_name** [string] The name of the organization the user belongs to.
- **organization_slug** [string] The slug of the organization the user belongs to.
- **organization_default_theme_id** [integer] The ID of the organizations's default theme.
- **created_at** [string/date-time] The date and time when the user was created.
- **sign_in_count** [integer] The number of times the user has signed in.
- **assuming_role** [boolean] Whether the user is assuming a role or not.
- **assuming_admin** [boolean] Whether the user is assuming admin.
- **assuming_admin_expiration** [string/date-time] When the user's admin role is set to expire.
- **superadmin_mode_expiration** [string/date-time] The user is in superadmin mode when set to a DateTime. The user is not in superadmin mode when set to null.
- **disable_non_compliant_fedramp_features** [boolean] Whether to disable non-compliant fedramp features.
- **created_by_id** [integer] The ID of the user who created this user.
- **last_updated_by_id** [integer] The ID of the user who last updated this user.

post(*name*, *email*, *primary_group_id*, *user*, ***, *active='DEFAULT'*, *city='DEFAULT'*, *state='DEFAULT'*, *time_zone='DEFAULT'*, *initials='DEFAULT'*, *department='DEFAULT'*, *title='DEFAULT'*, *prefers_sms_otp='DEFAULT'*, *group_ids='DEFAULT'*, *vpn_enabled='DEFAULT'*, *sso_disabled='DEFAULT'*, *otp_required_for_login='DEFAULT'*, *exempt_from_org_sms_otp_disabled='DEFAULT'*, *robot='DEFAULT'*, *send_email='DEFAULT'*)
    Create a new user (must be an admin or client user admin)
        Parameters
            name [string] The name of this user.
            email [string] The email of this user.
            primary_group_id [integer] The ID of the primary group of this user.
            user [string] The username of this user.
            active [boolean, optional] The account status of this user.
            city [string, optional] The city of this user.
            state [string, optional] The state of this user.

> > **time_zone** [string, optional] The time zone of this user.
> >
> > **initials** [string, optional] The initials of this user.
> >
> > **department** [string, optional] The department of this user.
> >
> > **title** [string, optional] The title of this user.
> >
> > **prefers_sms_otp** [boolean, optional] The preference for phone authorization of this
> > user
> >
> > **group_ids** [list, optional] An array of ids of all the groups this user is in.
> >
> > **vpn_enabled** [boolean, optional] The availability of vpn for this user.
> >
> > **sso_disabled** [boolean, optional] The availability of SSO for this user.
> >
> > **otp_required_for_login** [boolean, optional] The two factor authentication require-
> > ment for this user.
> >
> > **exempt_from_org_sms_otp_disabled** [boolean, optional] Whether the user has SMS
> > OTP enabled on an individual level. This field does not matter if the org does not
> > have SMS OTP disabled.
> >
> > **robot** [boolean, optional] Whether the user is a robot.
> >
> > **send_email** [boolean, optional] Whether the user will receive a welcome email.
>
> **Returns**
>
> > *civis.response.Response*
> >
> > > • **id** [integer] The ID of this user.
> > > • **user** [string] The username of this user.
> > > • **name** [string] The name of this user.
> > > • **email** [string] The email of this user.
> > > • **active** [boolean] The account status of this user.
> > > • **primary_group_id** [integer] The ID of the primary group of this user.
> > > • **groups** [list::] An array of all the groups this user is in. - id : integer
> > >
> > > > The ID of this group.
> > >
> > > > – **name** [string] The name of this group.
> > > >
> > > > – **organization_id** [integer] The ID of the organization associ-
> > > > ated with this group.
> > > >
> > > > – **organization_name** [string] The name of the organization
> > > > associated with this group.
> > >
> > > • **city** [string] The city of this user.
> > > • **state** [string] The state of this user.
> > > • **time_zone** [string] The time zone of this user.
> > > • **initials** [string] The initials of this user.
> > > • **department** [string] The department of this user.
> > > • **title** [string] The title of this user.
> > > • **github_username** [string] The GitHub username of this user.
> > > • **prefers_sms_otp** [boolean] The preference for phone authorization of this
> > > user
> > > • **vpn_enabled** [boolean] The availability of vpn for this user.
> > > • **sso_disabled** [boolean] The availability of SSO for this user.
> > > • **otp_required_for_login** [boolean] The two factor authentication require-
> > > ment for this user.
> > > • **exempt_from_org_sms_otp_disabled** [boolean] Whether the user has
> > > SMS OTP enabled on an individual level. This field does not matter
> > > if the org does not have SMS OTP disabled.
> > > • **sms_otp_allowed** [boolean] Whether the user is allowed to receive two
> > > factor authentication codes via SMS.
> > > • **robot** [boolean] Whether the user is a robot.
> > > • **phone** [string] The phone number of this user.

- **organization_slug** [string] The slug of the organization the user belongs
  to.
- **organization_sso_disable_capable** [boolean] The user's organization's
  ability to disable sso for their users.
- **organization_login_type** [string] The user's organization's login type.
- **organization_sms_otp_disabled** [boolean] Whether the user's organiza-
  tion has SMS OTP disabled.
- **my_permission_level** [string] Your permission level on the object. One
  of "read", "write", or "manage".
- **created_at** [string/date-time] The date and time when the user was cre-
  ated.
- **updated_at** [string/date-time] The date and time when the user was last
  updated.
- **last_seen_at** [string/date-time] The date and time when the user last vis-
  ited Platform.
- **suspended** [boolean] Whether the user is suspended due to inactivity.
- **created_by_id** [integer] The ID of the user who created this user.
- **last_updated_by_id** [integer] The ID of the user who last updated this
  user.

post_api_keys(*id*, *expires_in*, *name*, ***, *constraints='DEFAULT'*)

   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. - constraint
       : string

           The path matcher of the constraint.

   - **constraint_type** [string] The type of constraint (exact/prefix/regex/verb).
   - **get_allowed** [boolean] Whether the constraint allows GET requests.
   - **head_allowed** [boolean] Whether the constraint allows HEAD requests.
   - **post_allowed** [boolean] Whether the constraint allows POST requests.
   - **put_allowed** [boolean] Whether the constraint allows PUT requests.
   - **patch_allowed** [boolean] Whether the constraint allows PATCH requests.
   - **delete_allowed** [boolean] Whether the constraint allows DELETE re-
     quests.

   Returns

   civis.response.Response

   - **id** [integer] The ID of the API key.
   - **name** [string] The name of the API key.
   - **expires_at** [string/date-time] The date and time when the key expired.
   - **created_at** [string/date-time] The date and time when the key was created.
   - **revoked_at** [string/date-time] The date and time when the key was re-
     voked.
   - **last_used_at** [string/date-time] The date and time when the key was last
     used.
   - **scopes** [list] The scopes which the key is permissioned on.
   - **use_count** [integer] The number of times the key has been used.
   - **expired** [boolean] True if the key has expired.
   - **active** [boolean] True if the key has neither expired nor been revoked.
   - **constraints** [list::] Constraints on the abilities of the created key - con-
     straint : string

> > The path matcher of the constraint.
>
> > – **constraint_type** [string] The type of constraint (exact/prefix/regex/verb).
> >
> > – **get_allowed** [boolean] Whether the constraint allows GET requests.
> >
> > – **head_allowed** [boolean] Whether the constraint allows HEAD requests.
> >
> > – **post_allowed** [boolean] Whether the constraint allows POST requests.
> >
> > – **put_allowed** [boolean] Whether the constraint allows PUT requests.
> >
> > – **patch_allowed** [boolean] Whether the constraint allows PATCH requests.
> >
> > – **delete_allowed** [boolean] Whether the constraint allows DELETE requests.
>
> > • **token** [string] The API key.

post_me_favorites(*object_id*, *object_type*)

> Favorite an item
>
> > **Parameters**
> >
> > > **object_id** [integer] The id of the object. If specified as a query parameter, must also specify object_type parameter.
> > >
> > > **object_type** [string] The type of the object that is favorited. Valid options: Project
> >
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > • **id** [integer] The id of the favorite.
> > > • **object_id** [integer] The id of the object. If specified as a query parameter, must also specify object_type parameter.
> > > • **object_type** [string] The type of the object that is favorited. Valid options: Project
> > > • **object_name** [string] The name of the object that is favorited.
> > > • **created_at** [string/time] The time this favorite was created.

post_me_superadmin()

> Enables Superadmin Mode for the current user
>
> > **Returns**
> >
> > *civis.response.Response*
> >
> > > • **id** [integer] The ID of this user.
> > > • **name** [string] This user's name.
> > > • **email** [string] This user's email address.
> > > • **username** [string] This user's username.
> > > • **initials** [string] This user's initials.
> > > • **last_checked_announcements** [string/date-time] The date and time at which the user last checked their announcements.
> > > • **feature_flags** [dict] The feature flag settings for this user.
> > > • **roles** [list] The roles this user has, listed by slug.
> > > • **preferences** [dict] This user's preferences.
> > > • **custom_branding** [string] The branding of Platform for this user.
> > > • **primary_group_id** [integer] The ID of the primary group of this user.
> > > • **groups** [list::] An array of all the groups this user is in. - id : integer

The ID of this group.

- – **name** [string] The name of this group.
- – **organization_id** [integer] The ID of the organization associated with this group.
- – **organization_name** [string] The name of the organization associated with this group.
- **organization_name** [string] The name of the organization the user belongs to.
- **organization_slug** [string] The slug of the organization the user belongs to.
- **organization_default_theme_id** [integer] The ID of the organizations's default theme.
- **created_at** [string/date-time] The date and time when the user was created.
- **sign_in_count** [integer] The number of times the user has signed in.
- **assuming_role** [boolean] Whether the user is assuming a role or not.
- **assuming_admin** [boolean] Whether the user is assuming admin.
- **assuming_admin_expiration** [string/date-time] When the user's admin role is set to expire.
- **superadmin_mode_expiration** [string/date-time] The user is in superadmin mode when set to a DateTime. The user is not in superadmin mode when set to null.
- **disable_non_compliant_fedramp_features** [boolean] Whether to disable non-compliant fedramp features.
- **created_by_id** [integer] The ID of the user who created this user.
- **last_updated_by_id** [integer] The ID of the user who last updated this user.

post_unsuspend(*id*)

Unsuspends user

> **Parameters**
>> **id** [integer] The ID of this user.
>
> **Returns**
>> *civis.response.Response*
>>> - **id** [integer] The ID of this user.
>>> - **user** [string] The username of this user.
>>> - **unlocked_at** [string/date-time] The time the user's account was unsuspended

Workflows

class **Workflows**(*session_kwargs*, *client*, *return_type='civis'*)

Examples

```
>>> import civis
>>> client = civis.APIClient()
>>> client.workflows.list(...)
```

Methods

delete_projects(id, project_id)	Remove a Workflow 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 a Workflow
get_executions(id, execution_id)	Get a workflow execution
get_executions_tasks(id, execution_id, task_name)	Get a task of a workflow execution
get_git_commits(id, commit_hash)	Get file contents at git ref
list(*[, hidden, archived, author, state, ...])	List Workflows
list_dependencies(id, *[, user_id])	List dependent objects for this object
list_executions(id, *[, limit, page_num, ...])	List workflow executions
list_git(id)	Get the git metadata attached to an item
list_git_commits(id)	Get the git commits for an item on the current branch
list_projects(id, *[, hidden])	List the projects a Workflow belongs to
list_shares(id)	List users and groups permissioned on this object
patch(id, *[, name, description, ...])	Update some attributes of this Workflow
patch_git(id, *[, git_ref, git_branch, ...])	Update an attached git file
post(name, *[, description, from_job_chain, ...])	Create a Workflow
post_clone(id, *[, clone_schedule, ...])	Clone this Workflow
post_executions(id, *[, target_task, input, ...])	Execute a workflow
post_executions_cancel(id, execution_id)	Cancel a workflow execution
post_executions_resume(id, execution_id)	Resume a paused workflow execution
post_executions_retry(id, execution_id, *[, ...])	Retry a failed task, or all failed tasks in an execution
post_git_checkout(id)	Checkout content that the existing git_ref points to and save to the object
post_git_checkout_latest(id)	Checkout latest commit on the current branch of a script or workflow
post_git_commits(id, content, message, file_hash)	Commit and push a new version of the file
put(id, name, *[, description, definition, ...])	Replace all attributes of this Workflow
put_archive(id, status)	Update the archive status of this object
put_git(id, *[, git_ref, git_branch, ...])	Attach an item to a file in a git repo
put_projects(id, project_id)	Add a Workflow to a project
put_shares_groups(id, group_ids, ...[, ...])	Set the permissions groups has on this object
put_shares_users(id, user_ids, ...[, ...])	Set the permissions users have on this object
put_transfer(id, user_id, ...[, email_body, ...])	Transfer ownership of this object to another user

—

delete_projects(*id*, *project_id*)

> Remove a Workflow from a project
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the Workflow.
> > >
> > > **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] The ID of the resource that is shared.
> > >
> > > **group_id** [integer] The 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] The ID of the resource that is shared.
> > >
> > > **user_id** [integer] The ID of the user.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

get(*id*)

> Get a Workflow
>
> > **Parameters**
> >
> > > **id** [integer]
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **id** [integer] The ID for this workflow.
> > > > • **name** [string] The name of this workflow.
> > > > • **description** [string] A description of the workflow.
> > > > • **definition** [string] The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.
> > > > • **valid** [boolean] The validity of the workflow definition.
> > > > • **validation_errors** [string] The errors encountered when validating the workflow definition.
> > > > • **file_id** [string] The file id for the s3 file containing the workflow configuration.
> > > > • **user** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > • **state** [string] The state of the workflow. State is "running" if any execution is running, otherwise reflects most recent execution state.
> > > > • **schedule** [dict::]
> > > >
> > > > > – **scheduled** [boolean] If the item is scheduled.

- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

- **scheduled_hours** [list] Hours of the day it is scheduled on.

- **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **allow_concurrent_executions** [boolean] Whether the workflow can execute when already running.
- **time_zone** [string] The time zone of this workflow.
- **next_execution_at** [string/time] The time of the next scheduled execution.
- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on

    - **failure_on** [boolean] If failure email notifications are on
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- created_at : string/time
- updated_at : string/time

get_executions(*id*, *execution_id*)

Get a workflow execution

> **Parameters**
>
> > **id** [integer] The ID for the workflow.
> >
> > **execution_id** [integer] The ID for the workflow execution.
>
> **Returns**
>
> > *civis.response.Response*
> >
> > > - **id** [integer] The ID for this workflow execution.
> > > - **state** [string] The state of this workflow execution.
> > > - **mistral_state** [string] The state of this workflow as reported by mistral. One of running, paused, success, error, or cancelled
> > > - **mistral_state_info** [string] The state info of this workflow as reported by mistral.

- **user** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **definition** [string] The definition of the workflow for this execution.
- **input** [dict] Key-value pairs defined for this execution.
- **included_tasks** [list] The subset of workflow tasks selected to execute.
- **tasks** [list::] The tasks associated with this execution. - name : string

    The name of the task.

    - **mistral_state** [string] The state of this task. One of idle, waiting, running, delayed, success, error, or cancelled

    - **mistral_state_info** [string] Extra info associated with the state of the task.

    - **runs** [list::] The runs associated with this task, in descending order by id. - id : integer

        The ID of the run.

        * **job_id** [integer] The ID of the job associated with the run.

        * **state** [string] The state of the run.

    - **executions** [list::] The executions run by this task, in descending order by id. - id : integer

        The ID of the execution.

        * **workflow_id** [integer] The ID of the workflow associated with the execution.
- **started_at** [string/time] The time this execution started.
- **finished_at** [string/time] The time this execution finished.
- **created_at** [string/time] The time this execution was created.
- **updated_at** [string/time] The time this execution was last updated.

get_executions_tasks(*id*, *execution_id*, *task_name*)

> Get a task of a workflow execution
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the workflow.
> > >
> > > **execution_id** [integer] The ID for the workflow execution.
> > >
> > > **task_name** [string] The URL-encoded name of the task.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > - **name** [string] The name of the task.
> > > - **mistral_state** [string] The state of this task. One of idle, waiting, running, delayed, success, error, or cancelled
> > > - **mistral_state_info** [string] Extra info associated with the state of the task.

> - **runs** [list::] The runs associated with this task, in descending order by id.
>   - id : integer
>
>   The ID of the run.
>
>   - **job_id** [integer] The ID of the job associated with the run.
>
>   - **state** [string] The state of the run.
>
>   - **created_at** [string/time] The time that the run was queued.
>
>   - **started_at** [string/time] The time that the run started.
>
>   - **finished_at** [string/time] The time that the run completed.
>
> - **executions** [list::] The executions run by this task, in descending order by id. - id : integer
>
>   The ID of the execution.
>
>   - **workflow_id** [integer] The ID of the workflow associated with the execution.
>
>   - **state** [string] The state of this workflow execution.
>
>   - **created_at** [string/time] The time this execution was created.
>
>   - **started_at** [string/time] The time this execution started.
>
>   - **finished_at** [string/time] The time this execution finished.

get_git_commits(*id*, *commit_hash*)

> Get file contents at git ref
>
> **Parameters**
>> **id** [integer] The ID of the file.
>> **commit_hash** [string] The SHA (full or shortened) of the desired git commit.
>
> **Returns**
>> [*civis.response.Response*](#)
>>> - **content** [string] The file's contents.
>>> - **type** [string] The file's type.
>>> - **size** [integer] The file's size.
>>> - **file_hash** [string] The SHA of the file.

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

> List Workflows
>
> **Parameters**
>> **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
>> **archived** [string, optional] The archival status of the requested item(s).
>> **author** [string, optional] If specified, return items from any of these authors. It accepts a comma- separated list of user IDs.
>> **state** [array, optional] State of the most recent execution.One or more of queued, running, succeeded, failed, cancelled, idle, and scheduled.
>> **scheduled** [boolean, optional] If the workflow is scheduled.
>> **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 up-
dated_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

civis.response.PaginatedResponse

- **id** [integer] The ID for this workflow.
- **name** [string] The name of this workflow.
- **description** [string] A description of the workflow.
- **valid** [boolean] The validity of the workflow definition.
- **file_id** [string] The file id for the s3 file containing the workflow configu-
ration.
- **user** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

- **state** [string] The state of the workflow. State is "running" if any execution
is running, otherwise reflects most recent execution state.
- **schedule** [dict::]

    – **scheduled** [boolean] If the item is scheduled.

    – **scheduled_days** [list] Days of the week, based on numeric
value starting at 0 for Sunday. Mutually exclusive with
scheduledDaysOfMonth

    – **scheduled_hours** [list] Hours of the day it is scheduled on.

    – **scheduled_minutes** [list] Minutes of the day it is scheduled
on.

    – **scheduled_runs_per_hour** [integer] Alternative to sched-
uled minutes, number of times to run per hour.

    – **scheduled_days_of_month** [list] Days of the month it is
scheduled on, mutually exclusive with scheduledDays.

- **allow_concurrent_executions** [boolean] Whether the workflow can exe-
cute when already running.
- **time_zone** [string] The time zone of this workflow.
- **next_execution_at** [string/time] The time of the next scheduled execution.
- **archived** [string] The archival status of the requested item(s).
- created_at : string/time
- updated_at : string/time

list_dependencies(*id*, ***, *user_id='DEFAULT'*)

List dependent objects for this object

Parameters

id [integer] The ID of the resource that is shared.

user_id [integer, optional] ID of target user

Returns

> *civis.response.Response*
> - **object_type** [string] Dependent object type
> - **fco_type** [string] Human readable dependent object type
> - **id** [integer] Dependent object ID
> - **name** [string] Dependent object name, or nil if the requesting user cannot read this object
> - **permission_level** [string] Permission level of target user (not user's groups) for dependent object, or null if no target user
> - **shareable** [boolean] Whether or not the requesting user can share this object.

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

> List workflow executions
> > **Parameters**
> > > **id** [integer] The ID for this workflow.
> > > **limit** [integer, optional] Number of results to return. Defaults to 20. Maximum allowed is 50.
> > > **page_num** [integer, optional] Page number of the results to return. Defaults to the first page, 1.
> > > **order** [string, optional] The field on which to order the result set. Defaults to id. Must be one of: id, updated_at, created_at.
> > > **order_dir** [string, optional] Direction in which to sort, either asc (ascending) or desc (descending) defaulting to desc.
> > > **iterator** [bool, optional] If True, return a generator to iterate over all responses. Use when more results than the maximum allowed by limit are needed. When True, limit and page_num are ignored. Defaults to False.
> > **Returns**
> > > *civis.response.PaginatedResponse*
> > > - **id** [integer] The ID for this workflow execution.
> > > - **state** [string] The state of this workflow execution.
> > > - **mistral_state** [string] The state of this workflow as reported by mistral. One of running, paused, success, error, or cancelled
> > > - **mistral_state_info** [string] The state info of this workflow as reported by mistral.
> > > - **user** [dict::]
> > >
> > >   – **id** [integer] The ID of this user.
> > >
> > >   – **name** [string] This user's name.
> > >
> > >   – **username** [string] This user's username.
> > >
> > >   – **initials** [string] This user's initials.
> > >
> > >   – **online** [boolean] Whether this user is online.
> > > - **started_at** [string/time] The time this execution started.
> > > - **finished_at** [string/time] The time this execution finished.
> > > - **created_at** [string/time] The time this execution was created.
> > > - **updated_at** [string/time] The time this execution was last updated.

list_git(*id*)

> Get the git metadata attached to an item
> > **Parameters**
> > > **id** [integer] The ID of the file.
> > **Returns**
> > > *civis.response.Response*

- **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
- **git_branch** [string] The git branch that the file is on.
- **git_path** [string] The path of the file in the repository.
- **git_repo** [dict::]

    – **id** [integer] The ID for this git repository.

    – **repo_url** [string] The URL for this git repository.

    – created_at : string/time

    – updated_at : string/time
- **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
- **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

list_git_commits(*id*)

> Get the git commits for an item on the current branch
>> **Parameters**
>>> **id** [integer] The ID of the file.
>> **Returns**
>>> *civis.response.Response*
>>>> - **commit_hash** [string] The SHA of the commit.
>>>> - **author_name** [string] The name of the commit's author.
>>>> - **date** [string/time] The commit's timestamp.
>>>> - **message** [string] The commit message.

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

> List the projects a Workflow belongs to
>> **Parameters**
>>> **id** [integer] The ID of the Workflow.
>>> **hidden** [boolean, optional] If specified to be true, returns hidden items. Defaults to false, returning non-hidden items.
>> **Returns**
>>> *civis.response.Response*
>>>> - **id** [integer] The ID for this project.
>>>> - **author** [dict::]

    – **id** [integer] The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

    – **online** [boolean] Whether this user is online.

>>>> - **name** [string] The name of this project.
>>>> - **description** [string] A description of the project.
>>>> - **users** [list::] Users who can see the project. - id : integer

        The ID of this user.

    – **name** [string] This user's name.

    – **username** [string] This user's username.

    – **initials** [string] This user's initials.

> – **online** [boolean] Whether this user is online.
>
> - auto_share : boolean
> - created_at : string/time
> - updated_at : string/time
> - **archived** [string] The archival status of the requested item(s).

list_shares(*id*)

    List users and groups permissioned on this object

        Parameters

            id [integer] The ID of the resource that is shared.

        Returns

        [`civis.response.Response`](#)

> - **readers** [dict::]
>
>   > – **users** [list::]
>   >
>   >   > ∗ id : integer
>   >   >
>   >   > ∗ name : string
>   >
>   > – **groups** [list::]
>   >
>   >   > ∗ id : integer
>   >   >
>   >   > ∗ name : string
>
> - **writers** [dict::]
>
>   > – **users** [list::]
>   >
>   >   > ∗ id : integer
>   >   >
>   >   > ∗ name : string
>   >
>   > – **groups** [list::]
>   >
>   >   > ∗ id : integer
>   >   >
>   >   > ∗ name : string
>
> - **owners** [dict::]
>
>   > – **users** [list::]
>   >
>   >   > ∗ id : integer
>   >   >
>   >   > ∗ name : string
>   >
>   > – **groups** [list::]
>   >
>   >   > ∗ id : integer
>   >   >
>   >   > ∗ name : string
>
> - **total_user_shares** [integer] For owners, the number of total users shared. For writers and readers, the number of visible users shared.
> - **total_group_shares** [integer] For owners, the number of total groups shared. For writers and readers, the number of visible groups shared.

patch(*id*, ***, *name='DEFAULT'*, *description='DEFAULT'*, *definition='DEFAULT'*, *schedule='DEFAULT'*, *allow_concurrent_executions='DEFAULT'*, *time_zone='DEFAULT'*, *notifications='DEFAULT'*)

    Update some attributes of this Workflow

        Parameters

            id [integer] The ID for this workflow.

            name [string, optional] The name of this workflow.

            description [string, optional] A description of the workflow.

definition [string, optional] The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.

schedule [dict, optional::]

- **scheduled** [boolean] If the item is scheduled.
- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
- **scheduled_hours** [list] Hours of the day it is scheduled on.
- **scheduled_minutes** [list] Minutes of the day it is scheduled on.
- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

allow_concurrent_executions [boolean, optional] Whether the workflow can execute when already running.

time_zone [string, optional] The time zone of this workflow.

notifications [dict, optional::]

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

Returns

[*civis.response.Response*](civis.response.Response)

- **id** [integer] The ID for this workflow.
- **name** [string] The name of this workflow.
- **description** [string] A description of the workflow.
- **definition** [string] The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.
- **valid** [boolean] The validity of the workflow definition.
- **validation_errors** [string] The errors encountered when validating the workflow definition.
- **file_id** [string] The file id for the s3 file containing the workflow configuration.
- **user** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.

- **state** [string] The state of the workflow. State is "running" if any execution is running, otherwise reflects most recent execution state.
- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

- **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

- **scheduled_hours** [list] Hours of the day it is scheduled on.

- **scheduled_minutes** [list] Minutes of the day it is scheduled on.

- **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.

- **allow_concurrent_executions** [boolean] Whether the workflow can execute when already running.
- **time_zone** [string] The time zone of this workflow.
- **next_execution_at** [string/time] The time of the next scheduled execution.
- **notifications** [dict::]

  - **urls** [list] URLs to receive a POST request at job completion

  - **success_email_subject** [string] Custom subject line for success e-mail.

  - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

  - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

  - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

  - **success_on** [boolean] If success email notifications are on

  - **failure_on** [boolean] If failure email notifications are on
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- created_at : string/time
- updated_at : string/time

patch_git(*id*, ***, *git_ref='DEFAULT'*, *git_branch='DEFAULT'*, *git_path='DEFAULT'*,
  git_repo_url='DEFAULT', *git_ref_type='DEFAULT'*, *pull_from_git='DEFAULT'*)

Update an attached git file

Parameters

id [integer] The ID of the file.

git_ref [string, optional] A git reference specifying an unambiguous version of the file. Can be a branch name, or the full or shortened SHA of a commit.

git_branch [string, optional] The git branch that the file is on.

git_path [string, optional] The path of the file in the repository.

git_repo_url [string, optional] The URL of the git repository.

git_ref_type [string, optional] Specifies if the file is versioned by branch or tag.

pull_from_git [boolean, optional] Automatically pull latest commit from git. Only works for scripts.

> **Returns**
> *civis.response.Response*
> - **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
> - **git_branch** [string] The git branch that the file is on.
> - **git_path** [string] The path of the file in the repository.
> - **git_repo** [dict::]
>
>   - **id** [integer] The ID for this git repository.
>
>   - **repo_url** [string] The URL for this git repository.
>
>   - created_at : string/time
>
>   - updated_at : string/time
> - **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
> - **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

post(*name*, ***, *description='DEFAULT'*, *from_job_chain='DEFAULT'*, *definition='DEFAULT'*, *schedule='DEFAULT'*, *allow_concurrent_executions='DEFAULT'*, *time_zone='DEFAULT'*, *notifications='DEFAULT'*, *hidden='DEFAULT'*)

> Create a Workflow
> **Parameters**
> **name** [string] The name of this workflow.
> **description** [string, optional] A description of the workflow.
> **from_job_chain** [integer, optional] If specified, create a workflow from the job chain this job is in, and inherit the schedule from the root of the chain.
> **definition** [string, optional] The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.
> **schedule** [dict, optional::]
>   - **scheduled** [boolean] If the item is scheduled.
>   - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
>   - **scheduled_hours** [list] Hours of the day it is scheduled on.
>   - **scheduled_minutes** [list] Minutes of the day it is scheduled on.
>   - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
>   - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
> **allow_concurrent_executions** [boolean, optional] Whether the workflow can execute when already running.
> **time_zone** [string, optional] The time zone of this workflow.
> **notifications** [dict, optional::]
>   - **urls** [list] URLs to receive a POST request at job completion
>   - **success_email_subject** [string] Custom subject line for success e-mail.
>   - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.
>   - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.
>   - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.
>   - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

- **success_on** [boolean] If success email notifications are on
- **failure_on** [boolean] If failure email notifications are on

    hidden [boolean, optional] The hidden status of the item.

Returns

    civis.response.Response

- **id** [integer] The ID for this workflow.
- **name** [string] The name of this workflow.
- **description** [string] A description of the workflow.
- **definition** [string] The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.
- **valid** [boolean] The validity of the workflow definition.
- **validation_errors** [string] The errors encountered when validating the workflow definition.
- **file_id** [string] The file id for the s3 file containing the workflow configuration.
- **user** [dict::]

    - **id** [integer] The ID of this user.

    - **name** [string] This user's name.

    - **username** [string] This user's username.

    - **initials** [string] This user's initials.

    - **online** [boolean] Whether this user is online.
- **state** [string] The state of the workflow. State is "running" if any execution is running, otherwise reflects most recent execution state.
- **schedule** [dict::]

    - **scheduled** [boolean] If the item is scheduled.

    - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

    - **scheduled_hours** [list] Hours of the day it is scheduled on.

    - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

    - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

    - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **allow_concurrent_executions** [boolean] Whether the workflow can execute when already running.
- **time_zone** [string] The time zone of this workflow.
- **next_execution_at** [string/time] The time of the next scheduled execution.
- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

> > > > – **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.
> > > >
> > > > – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.
> > > >
> > > > – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.
> > > >
> > > > – **success_on** [boolean] If success email notifications are on
> > > >
> > > > – **failure_on** [boolean] If failure email notifications are on
> > >
> > > - **archived** [string] The archival status of the requested item(s).
> > > - **hidden** [boolean] The hidden status of the item.
> > > - **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
> > > - created_at : string/time
> > > - updated_at : string/time

post_clone(*id*, ***, *clone_schedule='DEFAULT'*, *clone_notifications='DEFAULT'*)

> Clone this Workflow
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the workflow.
> > >
> > > **clone_schedule** [boolean, optional] If true, also copy the schedule to the new workflow.
> > >
> > > **clone_notifications** [boolean, optional] If true, also copy the notifications to the new workflow.
> >
> > **Returns**
> >
> > > *civis.response.Response*
> > >
> > > > - **id** [integer] The ID for this workflow.
> > > > - **name** [string] The name of this workflow.
> > > > - **description** [string] A description of the workflow.
> > > > - **definition** [string] The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.
> > > > - **valid** [boolean] The validity of the workflow definition.
> > > > - **validation_errors** [string] The errors encountered when validating the workflow definition.
> > > > - **file_id** [string] The file id for the s3 file containing the workflow configuration.
> > > > - **user** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > - **state** [string] The state of the workflow. State is "running" if any execution is running, otherwise reflects most recent execution state.
> > > > - **schedule** [dict::]
> > > >
> > > > > – **scheduled** [boolean] If the item is scheduled.
> > > > >
> > > > > – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > > > >
> > > > > – **scheduled_hours** [list] Hours of the day it is scheduled on.

> > > – **scheduled_minutes** [list] Minutes of the day it is scheduled
> > > on.
> > >
> > > – **scheduled_runs_per_hour** [integer] Alternative to sched-
> > > uled minutes, number of times to run per hour.
> > >
> > > – **scheduled_days_of_month** [list] Days of the month it is
> > > scheduled on, mutually exclusive with scheduledDays.
> >
> > • **allow_concurrent_executions** [boolean] Whether the workflow can exe-
> > cute when already running.
> > • **time_zone** [string] The time zone of this workflow.
> > • **next_execution_at** [string/time] The time of the next scheduled execution.
> > • **notifications** [dict::]
> >
> > > – **urls** [list] URLs to receive a POST request at job completion
> > >
> > > – **success_email_subject** [string] Custom subject line for suc-
> > > cess e-mail.
> > >
> > > – **success_email_body** [string] Custom body text for success
> > > e-mail, written in Markdown.
> > >
> > > – **success_email_addresses** [list] Addresses to notify by e-mail
> > > when the job completes successfully.
> > >
> > > – **failure_email_addresses** [list] Addresses to notify by e-mail
> > > when the job fails.
> > >
> > > – **stall_warning_minutes** [integer] Stall warning emails will
> > > be sent after this amount of minutes.
> > >
> > > – **success_on** [boolean] If success email notifications are on
> > >
> > > – **failure_on** [boolean] If failure email notifications are on
> >
> > • **archived** [string] The archival status of the requested item(s).
> > • **hidden** [boolean] The hidden status of the item.
> > • **my_permission_level** [string] Your permission level on the object. One
> > of "read", "write", or "manage".
> > • created_at : string/time
> > • updated_at : string/time

post_executions(*id*, ***, *target_task='DEFAULT'*, *input='DEFAULT'*, *included_tasks='DEFAULT'*)
    Execute a workflow
        Parameters
            id [integer] The ID for the workflow.
            target_task [string, optional] For a reverse workflow, the name of the task to target.
            input [dict, optional] Key-value pairs to send to this execution as inputs.
            included_tasks [list, optional] If specified, executes only the subset of workflow tasks
                included as specified by task name.
        Returns
            civis.response.Response
> > • **id** [integer] The ID for this workflow execution.
> > • **state** [string] The state of this workflow execution.
> > • **mistral_state** [string] The state of this workflow as reported by mistral.
> > One of running, paused, success, error, or cancelled
> > • **mistral_state_info** [string] The state info of this workflow as reported by
> > mistral.
> > • **user** [dict::]
> >
> > > – **id** [integer] The ID of this user.

- **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.

- **definition** [string] The definition of the workflow for this execution.
- **input** [dict] Key-value pairs defined for this execution.
- **included_tasks** [list] The subset of workflow tasks selected to execute.
- **tasks** [list::] The tasks associated with this execution. - name : string

  The name of the task.

  - **mistral_state** [string] The state of this task. One of idle, waiting, running, delayed, success, error, or cancelled

  - **mistral_state_info** [string] Extra info associated with the state of the task.

  - **runs** [list::] The runs associated with this task, in descending order by id. - id : integer

    The ID of the run.

    * **job_id** [integer] The ID of the job associated with the run.

    * **state** [string] The state of the run.

  - **executions** [list::] The executions run by this task, in descending order by id. - id : integer

    The ID of the execution.

    * **workflow_id** [integer] The ID of the workflow associated with the execution.

- **started_at** [string/time] The time this execution started.
- **finished_at** [string/time] The time this execution finished.
- **created_at** [string/time] The time this execution was created.
- **updated_at** [string/time] The time this execution was last updated.

post_executions_cancel(*id*, *execution_id*)

Cancel a workflow execution

Parameters

id [integer] The ID for the workflow.

execution_id [integer] The ID for the workflow execution.

Returns

[*civis.response.Response*](#)

- **id** [integer] The ID for this workflow execution.
- **state** [string] The state of this workflow execution.
- **mistral_state** [string] The state of this workflow as reported by mistral. One of running, paused, success, error, or cancelled
- **mistral_state_info** [string] The state info of this workflow as reported by mistral.
- **user** [dict::]

  - **id** [integer] The ID of this user.

> > > > – **name** [string] This user's name.
> > > >
> > > > – **username** [string] This user's username.
> > > >
> > > > – **initials** [string] This user's initials.
> > > >
> > > > – **online** [boolean] Whether this user is online.
> > >
> > > - **definition** [string] The definition of the workflow for this execution.
> > > - **input** [dict] Key-value pairs defined for this execution.
> > > - **included_tasks** [list] The subset of workflow tasks selected to execute.
> > > - **tasks** [list::] The tasks associated with this execution. - name : string
> > >
> > > > The name of the task.
> > > >
> > > > – **mistral_state** [string] The state of this task. One of idle, waiting, running, delayed, success, error, or cancelled
> > > >
> > > > – **mistral_state_info** [string] Extra info associated with the state of the task.
> > > >
> > > > – **runs** [list::] The runs associated with this task, in descending order by id. - id : integer
> > > >
> > > > > The ID of the run.
> > > > >
> > > > > ∗ **job_id** [integer] The ID of the job associated with the run.
> > > > >
> > > > > ∗ **state** [string] The state of the run.
> > > >
> > > > – **executions** [list::] The executions run by this task, in descending order by id. - id : integer
> > > >
> > > > > The ID of the execution.
> > > > >
> > > > > ∗ **workflow_id** [integer] The ID of the workflow associated with the execution.
> > >
> > > - **started_at** [string/time] The time this execution started.
> > > - **finished_at** [string/time] The time this execution finished.
> > > - **created_at** [string/time] The time this execution was created.
> > > - **updated_at** [string/time] The time this execution was last updated.

post_executions_resume(*id*, *execution_id*)

> Resume a paused workflow execution
>
> > **Parameters**
> >
> > > **id** [integer] The ID for the workflow.
> > >
> > > **execution_id** [integer] The ID for the workflow execution.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > - **id** [integer] The ID for this workflow execution.
> > > > - **state** [string] The state of this workflow execution.
> > > > - **mistral_state** [string] The state of this workflow as reported by mistral. One of running, paused, success, error, or cancelled
> > > > - **mistral_state_info** [string] The state info of this workflow as reported by mistral.
> > > > - **user** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.

- **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **definition** [string] The definition of the workflow for this execution.
- **input** [dict] Key-value pairs defined for this execution.
- **included_tasks** [list] The subset of workflow tasks selected to execute.
- **tasks** [list::] The tasks associated with this execution. - name : string

   The name of the task.

   - **mistral_state** [string] The state of this task. One of idle, waiting, running, delayed, success, error, or cancelled

   - **mistral_state_info** [string] Extra info associated with the state of the task.

   - **runs** [list::] The runs associated with this task, in descending order by id. - id : integer

      The ID of the run.

      * **job_id** [integer] The ID of the job associated with the run.

      * **state** [string] The state of the run.

   - **executions** [list::] The executions run by this task, in descending order by id. - id : integer

      The ID of the execution.

      * **workflow_id** [integer] The ID of the workflow associated with the execution.

- **started_at** [string/time] The time this execution started.
- **finished_at** [string/time] The time this execution finished.
- **created_at** [string/time] The time this execution was created.
- **updated_at** [string/time] The time this execution was last updated.

post_executions_retry(*id*, *execution_id*, ***, *task_name='DEFAULT'*)

   Retry a failed task, or all failed tasks in an execution

   Parameters

   id [integer] The ID for the workflow.

   execution_id [integer] The ID for the workflow execution.

   task_name [string, optional] If specified, the name of the task to be retried. If not specified, all failed tasks in the execution will be retried.

   Returns

   civis.response.Response

   - **id** [integer] The ID for this workflow execution.
   - **state** [string] The state of this workflow execution.
   - **mistral_state** [string] The state of this workflow as reported by mistral. One of running, paused, success, error, or cancelled
   - **mistral_state_info** [string] The state info of this workflow as reported by mistral.
   - **user** [dict::]

- **id** [integer] The ID of this user.

- **name** [string] This user's name.

- **username** [string] This user's username.

- **initials** [string] This user's initials.

- **online** [boolean] Whether this user is online.
- **definition** [string] The definition of the workflow for this execution.
- **input** [dict] Key-value pairs defined for this execution.
- **included_tasks** [list] The subset of workflow tasks selected to execute.
- **tasks** [list::] The tasks associated with this execution. - name : string

  The name of the task.

  - **mistral_state** [string] The state of this task. One of idle, waiting, running, delayed, success, error, or cancelled

  - **mistral_state_info** [string] Extra info associated with the state of the task.

  - **runs** [list::] The runs associated with this task, in descending order by id. - id : integer

    The ID of the run.

    * **job_id** [integer] The ID of the job associated with the run.

    * **state** [string] The state of the run.

  - **executions** [list::] The executions run by this task, in descending order by id. - id : integer

    The ID of the execution.

    * **workflow_id** [integer] The ID of the workflow associated with the execution.

- **started_at** [string/time] The time this execution started.
- **finished_at** [string/time] The time this execution finished.
- **created_at** [string/time] The time this execution was created.
- **updated_at** [string/time] The time this execution was last updated.

post_git_checkout(*id*)

    Checkout content that the existing git_ref points to and save to the object

        Parameters

            id [integer] The ID of the file.

        Returns

            civis.response.Response

- **content** [string] The file's contents.
- **type** [string] The file's type.
- **size** [integer] The file's size.
- **file_hash** [string] The SHA of the file.

post_git_checkout_latest(*id*)

    Checkout latest commit on the current branch of a script or workflow

        Parameters

            id [integer] The ID of the file.

> **Returns**
>> [`civis.response.Response`](#)
>>> - **content** [string] The file's contents.
>>> - **type** [string] The file's type.
>>> - **size** [integer] The file's size.
>>> - **file_hash** [string] The SHA of the file.

post_git_commits(*id*, *content*, *message*, *file_hash*)

> Commit and push a new version of the file
>> **Parameters**
>>> **id** [integer] The ID of the file.
>>> **content** [string] The contents to commit to the file.
>>> **message** [string] A commit message describing the changes being made.
>>> **file_hash** [string] The full SHA of the file being replaced.
>> **Returns**
>>> [`civis.response.Response`](#)
>>>> - **content** [string] The file's contents.
>>>> - **type** [string] The file's type.
>>>> - **size** [integer] The file's size.
>>>> - **file_hash** [string] The SHA of the file.

put(*id*, *name*, *, *description='DEFAULT'*, *definition='DEFAULT'*, *schedule='DEFAULT'*, *allow_concurrent_executions='DEFAULT'*, *time_zone='DEFAULT'*, *notifications='DEFAULT'*)

> Replace all attributes of this Workflow
>> **Parameters**
>>> **id** [integer] The ID for this workflow.
>>> **name** [string] The name of this workflow.
>>> **description** [string, optional] A description of the workflow.
>>> **definition** [string, optional] The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.
>>> **schedule** [dict, optional::]
>>>> - **scheduled** [boolean] If the item is scheduled.
>>>> - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
>>>> - **scheduled_hours** [list] Hours of the day it is scheduled on.
>>>> - **scheduled_minutes** [list] Minutes of the day it is scheduled on.
>>>> - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.
>>>> - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
>>> **allow_concurrent_executions** [boolean, optional] Whether the workflow can execute when already running.
>>> **time_zone** [string, optional] The time zone of this workflow.
>>> **notifications** [dict, optional::]
>>>> - **urls** [list] URLs to receive a POST request at job completion
>>>> - **success_email_subject** [string] Custom subject line for success e-mail.
>>>> - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.
>>>> - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.
>>>> - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.
>>>> - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.
>>>> - **success_on** [boolean] If success email notifications are on

- **failure_on** [boolean] If failure email notifications are on

Returns

[*civis.response.Response*](#)

- **id** [integer] The ID for this workflow.
- **name** [string] The name of this workflow.
- **description** [string] A description of the workflow.
- **definition** [string] The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.
- **valid** [boolean] The validity of the workflow definition.
- **validation_errors** [string] The errors encountered when validating the workflow definition.
- **file_id** [string] The file id for the s3 file containing the workflow configuration.
- **user** [dict::]

  - **id** [integer] The ID of this user.

  - **name** [string] This user's name.

  - **username** [string] This user's username.

  - **initials** [string] This user's initials.

  - **online** [boolean] Whether this user is online.
- **state** [string] The state of the workflow. State is "running" if any execution is running, otherwise reflects most recent execution state.
- **schedule** [dict::]

  - **scheduled** [boolean] If the item is scheduled.

  - **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth

  - **scheduled_hours** [list] Hours of the day it is scheduled on.

  - **scheduled_minutes** [list] Minutes of the day it is scheduled on.

  - **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

  - **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **allow_concurrent_executions** [boolean] Whether the workflow can execute when already running.
- **time_zone** [string] The time zone of this workflow.
- **next_execution_at** [string/time] The time of the next scheduled execution.
- **notifications** [dict::]

  - **urls** [list] URLs to receive a POST request at job completion

  - **success_email_subject** [string] Custom subject line for success e-mail.

  - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

  - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

> > > > – **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.
> > > >
> > > > – **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.
> > > >
> > > > – **success_on** [boolean] If success email notifications are on
> > > >
> > > > – **failure_on** [boolean] If failure email notifications are on
> > >
> > > • **archived** [string] The archival status of the requested item(s).
> > > • **hidden** [boolean] The hidden status of the item.
> > > • **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
> > > • created_at : string/time
> > > • updated_at : string/time

put_archive(*id*, *status*)

> Update the archive status of this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the object.
> > > **status** [boolean] The desired archived status of the object.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **id** [integer] The ID for this workflow.
> > > > • **name** [string] The name of this workflow.
> > > > • **description** [string] A description of the workflow.
> > > > • **definition** [string] The definition of the workflow in YAML format. Must not be specified if *fromJobChain* is specified.
> > > > • **valid** [boolean] The validity of the workflow definition.
> > > > • **validation_errors** [string] The errors encountered when validating the workflow definition.
> > > > • **file_id** [string] The file id for the s3 file containing the workflow configuration.
> > > > • **user** [dict::]
> > > >
> > > > > – **id** [integer] The ID of this user.
> > > > >
> > > > > – **name** [string] This user's name.
> > > > >
> > > > > – **username** [string] This user's username.
> > > > >
> > > > > – **initials** [string] This user's initials.
> > > > >
> > > > > – **online** [boolean] Whether this user is online.
> > > >
> > > > • **state** [string] The state of the workflow. State is "running" if any execution is running, otherwise reflects most recent execution state.
> > > > • **schedule** [dict::]
> > > >
> > > > > – **scheduled** [boolean] If the item is scheduled.
> > > > >
> > > > > – **scheduled_days** [list] Days of the week, based on numeric value starting at 0 for Sunday. Mutually exclusive with scheduledDaysOfMonth
> > > > >
> > > > > – **scheduled_hours** [list] Hours of the day it is scheduled on.
> > > > >
> > > > > – **scheduled_minutes** [list] Minutes of the day it is scheduled on.
> > > > >
> > > > > – **scheduled_runs_per_hour** [integer] Alternative to scheduled minutes, number of times to run per hour.

- **scheduled_days_of_month** [list] Days of the month it is scheduled on, mutually exclusive with scheduledDays.
- **allow_concurrent_executions** [boolean] Whether the workflow can execute when already running.
- **time_zone** [string] The time zone of this workflow.
- **next_execution_at** [string/time] The time of the next scheduled execution.
- **notifications** [dict::]

    - **urls** [list] URLs to receive a POST request at job completion

    - **success_email_subject** [string] Custom subject line for success e-mail.

    - **success_email_body** [string] Custom body text for success e-mail, written in Markdown.

    - **success_email_addresses** [list] Addresses to notify by e-mail when the job completes successfully.

    - **failure_email_addresses** [list] Addresses to notify by e-mail when the job fails.

    - **stall_warning_minutes** [integer] Stall warning emails will be sent after this amount of minutes.

    - **success_on** [boolean] If success email notifications are on

    - **failure_on** [boolean] If failure email notifications are on
- **archived** [string] The archival status of the requested item(s).
- **hidden** [boolean] The hidden status of the item.
- **my_permission_level** [string] Your permission level on the object. One of "read", "write", or "manage".
- created_at : string/time
- updated_at : string/time

put_git(*id*, *, *git_ref='DEFAULT'*, *git_branch='DEFAULT'*, *git_path='DEFAULT'*, *git_repo_url='DEFAULT'*, *git_ref_type='DEFAULT'*, *pull_from_git='DEFAULT'*)

Attach an item to a file in a git repo

Parameters

id [integer] The ID of the file.

git_ref [string, optional] A git reference specifying an unambiguous version of the file. Can be a branch name, or the full or shortened SHA of a commit.

git_branch [string, optional] The git branch that the file is on.

git_path [string, optional] The path of the file in the repository.

git_repo_url [string, optional] The URL of the git repository.

git_ref_type [string, optional] Specifies if the file is versioned by branch or tag.

pull_from_git [boolean, optional] Automatically pull latest commit from git. Only works for scripts.

Returns

civis.response.Response

- **git_ref** [string] A git reference specifying an unambiguous version of the file. Can be a branch name, tag or the full or shortened SHA of a commit.
- **git_branch** [string] The git branch that the file is on.
- **git_path** [string] The path of the file in the repository.
- **git_repo** [dict::]

    - **id** [integer] The ID for this git repository.

> > > > > > – **repo_url** [string] The URL for this git repository.
> > > > > >
> > > > > > – created_at : string/time
> > > > > >
> > > > > > – updated_at : string/time
> > > > >
> > > > > • **git_ref_type** [string] Specifies if the file is versioned by branch or tag.
> > > > > • **pull_from_git** [boolean] Automatically pull latest commit from git. Only works for scripts and workflows (assuming you have the feature enabled)

put_projects(*id*, *project_id*)

> Add a Workflow to a project
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the Workflow.
> > > **project_id** [integer] The ID of the project.
> >
> > **Returns**
> >
> > > **None** Response code 204: success

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

> Set the permissions groups has on this object
>
> > **Parameters**
> >
> > > **id** [integer] The ID of the resource that is shared.
> > > **group_ids** [list] An array of one or more group IDs.
> > > **permission_level** [string] Options are: "read", "write", or "manage".
> > > **share_email_body** [string, optional] Custom body text for e-mail sent on a share.
> > > **send_shared_email** [boolean, optional] Send email to the recipients of a share.
> >
> > **Returns**
> >
> > > [*civis.response.Response*](#)
> > >
> > > > • **readers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **writers** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > >
> > > > • **owners** [dict::]
> > > >
> > > > > – **users** [list::]
> > > > >
> > > > > > ∗ id : integer
> > > > > >
> > > > > > ∗ name : string
> > > > >
> > > > > – **groups** [list::]

           ∗ id : integer

           ∗ name : string

       • **total_user_shares** [integer] For owners, the number of total users shared.
         For writers and readers, the number of visible users shared.

       • **total_group_shares** [integer] For owners, the number of total groups
         shared. For writers and readers, the number of visible groups shared.

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

  Set the permissions users have on this object

    Parameters

      id [integer] The ID of the resource that is shared.

      user_ids [list] An array of one or more user IDs.

      permission_level [string] Options are: "read", "write", or "manage".

      share_email_body [string, optional] Custom body text for e-mail sent on a share.

      send_shared_email [boolean, optional] Send email to the recipients of a share.

    Returns

    [*civis.response.Response*](#)

       • **readers** [dict::]

          – **users** [list::]

             ∗ id : integer

             ∗ name : string

          – **groups** [list::]

             ∗ id : integer

             ∗ name : string

       • **writers** [dict::]

          – **users** [list::]

             ∗ id : integer

             ∗ name : string

          – **groups** [list::]

             ∗ id : integer

             ∗ name : string

       • **owners** [dict::]

          – **users** [list::]

             ∗ id : integer

             ∗ name : string

          – **groups** [list::]

             ∗ id : integer

             ∗ name : string

       • **total_user_shares** [integer] For owners, the number of total users shared.
         For writers and readers, the number of visible users shared.

       • **total_group_shares** [integer] For owners, the number of total groups
         shared. For writers and readers, the number of visible groups shared.

put_transfer(*id*, *user_id*, *include_dependencies*, ***, *email_body='DEFAULT'*, *send_email='DEFAULT'*)
   Transfer ownership of this object to another user

   Parameters

   id [integer] The ID of the resource that is shared.

   user_id [integer] ID of target user

   include_dependencies [boolean] Whether or not to give manage permissions on all
      dependencies

   email_body [string, optional] Custom body text for e-mail sent on transfer.

   send_email [boolean, optional] Send email to the target user of the transfer?

   Returns

   civis.response.Response

   - **dependencies** [list::] Dependent objects for this object - object_type :
      string

         Dependent object type

         – **fco_type** [string] Human readable dependent object type

         – **id** [integer] Dependent object ID

         – **name** [string] Dependent object name, or nil if the requesting
            user cannot read this object

         – **permission_level** [string] Permission level of target user (not
            user's groups) for dependent object, or null if no target
            user

         – **shared** [boolean] Whether dependent object was successfully
            shared with target user

5.6 Command Line Interface

A command line interface (CLI) to Civis is provided. This can be invoked by typing the command `civis` in the shell
(sh, bash, zsh, etc.). It can also be used in Civis container scripts where the Docker image has this client installed.
Here's a simple example of printing the types of scripts.

```
> civis scripts list-types
- name: sql
- name: python3
- name: javascript
- name: r
- name: containers
```

Not all API endpoints are available through the CLI since some take complex data types (e.g., arrays, objects/dictionaries) as input. However, functionality is available for getting information about scripts, logs, etc., as
well as executing already created scripts.

There are a few extra, CLI-only commands that wrap the Files API endpoints to make uploading and downloading files
easier: `civis files upload $PATH` and `civis files download $FILEID $PATH`.

The default output format is YAML, but the `--json-output` allows you to get output in JSON.

You can find out more information about a command by adding a `--help` option, like `civis scripts list --help`.

5.6.1 Job Logs

These commands show job run logs in the format: "datetime message\n" where datetime is in ISO8601 format, like "2020-02-14T20:28:18.722Z". If the job is still running, this command will continue outputting logs until the run is done and then exit. If the run is already finished, it will output all the logs from that run and then exit.

NOTE: These commands could miss some log entries from a currently-running job. It does not re-fetch logs that might have been saved out of order, to preserve the chronological order of the logs and without duplication.

- `civis jobs follow-log $JOB_ID`

  Output live log from the most recent job run for the given job ID.

- `civis jobs follow-run-log $JOB_ID $RUN_ID`

  Output live log from the given job and run ID.

5.6.2 Notebooks

The following CLI-only commands make it easier to use Civis Platform as a backend for your Jupyter notebooks.

- `civis notebooks download $NOTEBOOK_ID $PATH`

  Download a notebook from Civis Platform to the requested file on the local filesystem.

- `civis notebooks new [$LANGUAGE] [--mem $MEMORY] [--cpu $CPU]`

  Create a new notebook, allocate resources for it, and open it in a tab of your default web browser. This command is the most similar to `jupyter notebook`. By default, Civis Platform will create a Python 3 notebook, but you can request any other language. Optional resource parameters let you allocate more memory or CPU to your notebook.

- `civis notebooks up $NOTEBOOK_ID [--mem $MEMORY] [--cpu $CPU]`

  Allocate resources for a notebook which already exists in Civis Platform and open it in a tab of your default browser. Optional resource arguments allow you to change resources allocated to your notebook (default to using the same resources as the previous run).

- `civis notebooks down $NOTEBOOK_ID`

  Stop a running notebook and free up the resources allocated to it.

- `civis notebooks open $NOTEBOOK_ID`

  Open an existing notebook (which may or may not be running) in your default browser.

5.6.3 SQL

The Civis CLI allows for easy running of SQL queries on Civis Platform through the following commands:

- `civis sql [-n $MAX_LINES] -d $DATABASE_NAME -f $FILE_NAME`

  Read a SQL query from a text file and run it on the specified database. The results of the query, if any, will be shown after it completes (up to a maximum of $MAX_LINES rows, defaulting to 100).

- `civis sql [-n $MAX_LINES] -d $DATABASE_NAME -c [$SQL_QUERY]`

  Instead of reading from a file, read query text from a command line argument. If you do not provide a query on the command line, the query text will be taken from stdin.

- `civis sql -d $DATABASE_NAME [-f $SQL_FILE_NAME] -o $OUTPUT_FILE_NAME`

With the *-o* or *–output* option specified, the complete results of the query will be downloaded to a CSV file at the requested location after the query completes.

5.7 Running Jobs and Templates

The `civis.utils` namespace provides several functions for running jobs and templates on the Civis Platform.

run_job(job_id[, api_key, client, ...])	Run a job.
run_template(id, arguments[, JSONValue, client])	Run a template and return the results.

5.7.1 civis.utils.run_job

civis.utils.**run_job**(*job_id*, *api_key=None*, *client=None*, *polling_interval=None*)

> Run a job.
>
> > **Parameters**
> >
> > > **job_id: str or int** The ID of the job.
> > >
> > > **api_key: DEPRECATED str, optional** Your Civis API key. If not given, the `CIVIS_API_KEY` environment variable will be used.
> > >
> > > **client: :class:`civis.APIClient`, optional** If not provided, an `civis.APIClient` object will be created from the `CIVIS_API_KEY`.
> > >
> > > **polling_interval** [int or float, optional] The number of seconds between API requests to check whether a result is ready.
> >
> > **Returns**
> >
> > > **results: *CivisFuture*** A *CivisFuture* object.

5.7.2 civis.utils.run_template

civis.utils.**run_template**(*id*, *arguments*, *JSONValue=False*, *client=None*)

> Run a template and return the results.
>
> > **Parameters**
> >
> > > **id: int** The template id to be run.
> > >
> > > **arguments: dict** Dictionary of arguments to be passed to the template.
> > >
> > > **JSONValue: bool, optional** If True, will return the JSON output of the template. If False, will return the file ids associated with the output results.
> > >
> > > **client: :class:`civis.APIClient`, optional** If not provided, an `civis.APIClient` object will be created from the `CIVIS_API_KEY`.
> >
> > **Returns**
> >
> > > **output: dict** If JSONValue = False, dictionary of file ids with the keys being their output names. If JSONValue = True, JSON dict containing the results of the template run. Expects only a single JSON result. Will return nothing if either there is no JSON result or there is more than 1 JSON result.

Examples

```
>>> # Run template to return file_ids
>>> run_template(my_template_id, arguments=my_dict_of_args)
{'output': 1234567}
>>> # Run template to return JSON output
>>> run_template(my_template_id, arguments=my_dict_of_args, JSONValue=True)
{'result1': 'aaa', 'result2': 123}
```

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

C

Q

R

S

T

U

W