
FRED Command Line Interface

['Epistemix Inc.']

Oct 25, 2021

CONTENTS

1	User Guide	3
1.1	Batch Configuration Files	3
1.2	Batch Execution Scripts	3
2	FRED Command-Line Interface Reference	5
2.1	fredcli	5
2.2	fredcli batch	6
2.3	fredcli env	19
2.4	fredcli user	21
	Index	23

Return to docs home

As part of the FRED Modeling Platform™ provided by [Epistemix](#), a command-line interface is provided. This FRED command-line interface, or `fredcli`, provides console-based access to FRED Cloud for the simulation and management of FRED programs.

If you have questions about Epistemix or are interested in speaking with our sales, recruiting, or other teams, please visit our main site at epistemix.com.

This document is a work in progress, so some areas may be light on content. The reference pages should be up to date, and we would recommend starting there.

USER GUIDE

The User Guide will provide examples showing how to use FREDpy. It will be developed more fully once we have settled on common usage patterns and developed the code base.

The following sections need to be added here:

- Installation
- User Guide

In the meantime, the following sections are currently available.

1.1 Batch Configuration Files

This section discusses the format of configuration files used to submit batches to FRED Cloud.

1.2 Batch Execution Scripts

This section discusses the format of the execution script used to submit batches to FRED Cloud.

FRED COMMAND-LINE INTERFACE REFERENCE

This provides reference information for the `fredcli` commands.

2.1 `fredcli`

The unified command-line tool to manage FRED programs in the cloud.

2.1.1 Synopsis

`fredcli command`

CLI tool to manage full development cycle of projects

```
fredcli command [options] <command>
```

Commands

`batch`

These are the batch functions for `fredcli`.

`env`

These are the environment functions for...

`login`

[Deprecated, please see `fredcli user`...]

`user`

These are the user commands

2.1.2 Description

Replace with a description

2.1.3 Available Commands

- *fredcli batch*
- *fredcli env*
- *fredcli user*

2.2 fredcli batch

The `fredcli batch` commands allow a modeler to submit and interact with batches in FRED Cloud.

2.2.1 Synopsis

fredcli batch

These are the batch functions for `fredcli`.

```
fredcli batch [options] <subcommand>
```

Commands

get-logs

Display the most recent log events for the...

get-results

Downloads the results associated with...

get-status

Gets the current status of a batch.

get-status

Gets the current status of a batch.

list

Display data about the most recent 20...

submit

Execute the model in the current directory...

validate-config

Validate the structure of a batch...

2.2.2 Description

The `fredcli batch` commands perform various operations on batches. A batch represents an request to perform work in FRED Cloud, typically to perform one or more simulations of a FRED program. Each batch has a unique identifier that identifies the batch.

See the available commands for details on the available actions.

2.2.3 Available Commands

fredcli batch get-logs

Display log events for a batch.

Synopsis

fredcli batch get-logs command

Display the most recent log events for the given batch.

```
fredcli batch get-logs command [options] BATCH_ID
```

Options

-l, --limit <limit>

The maximum number of log events to retrieve.

Default 10

-i, --interval <interval>

The amount of time, in hours (h) or days (d), in which to find events. For example, **-i 24h** (24 hours) or **-i 2d** (2 days).

Default 12h

--output <output>

The output format to use.

Default text

Options text | json | yaml

-v, --verbose

Show more detailed output.

Arguments

BATCH_ID

Required argument

Description

This command retrieves log events from FRED Cloud for a given batch and displays them in the console. By default, up to 10 events within the past 12 hours are shown. The **-l / --limit** option can be used to specify how many events to show, and the **-i / --interval** option can specify how far back in time to look for events.

The output for this command can be text, json, or yaml, as specified by the **--output** option. For non-text output, more detailed information is shown if the **-v / --verbose** flag is given. Each event includes the date and GMT time for when the event occurred.

The specific events shown are dependent on the type of batch and the specific version of FRED used.

Examples

The following example shows the logs from an in-process batch simulating events in the Chicago, IL area.

```
fredcli batch get-logs 7bd3ede8a23d474b82c021a7e9cf67eb
2021-06-18 07:19:15.080000 GMT+0000 run_set 1 completed at Fri Jun 18 07:19:15 GMT 2021
2021-06-18 07:19:15.080000 GMT+0000
2021-06-18 07:19:15.083000 GMT+0000 fred_job: running job chicago-event id 1 run 11 ...
2021-06-18 07:19:15.085000 GMT+0000 fred_job: running job chicago-event id 1 run 12 ...
2021-06-18 07:19:15.088000 GMT+0000 fred_job: running job chicago-event id 1 run 13 ...
2021-06-18 07:19:15.092000 GMT+0000 fred_job: running job chicago-event id 1 run 14 ...
2021-06-18 07:19:15.094000 GMT+0000 fred_job: running job chicago-event id 1 run 15 ...
```

Here is another call for the same simulation using JSON output and the `--verbose` flag. Only a partial output is shown.

```
$ fredcli batch get-logs 7bd3ede8a23d474b82c021a7e9cf67eb --output json -v
[
  . . .
  {
    "timestamp": "2021-06-18 07:19:15.083000 GMT+0000",
    "message": {
      "docker": {
        "container_id":
↪ "333dae8f08db87b585b08b8698c11c1f17ef80ec805cd1f03a7ad462ab642da8"
      },
      "kubernetes": {
        "container_hash": "335566905560.dkr.ecr.us-east-1.amazonaws.com/fred-cloud-
↪ runner@sha256:34b0ae0d1aab339c6a8ad15b252ac9ab2365e835ad672e392e7a19c742c23d58",
        "container_image": "335566905560.dkr.ecr.us-east-1.amazonaws.com/fred-cloud-
↪ runner:latest",
        "container_name": "fred-runner",
        "host": "ip-192-168-2-38.ec2.internal",
        "labels": {
          "app": "fred-cloud",
          "controller-uid": "282b7777-7540-4219-b612-b49af669a351",
          "job-name": "fred-job-7bd3ede8a23d474b82c021a7e9cf67eb"
        },
        "namespace_name": "fred-runner",
        "pod_id": "57c2c671-0801-4672-b484-e3c796d34f3e",
        "pod_name": "fred-job-7bd3ede8a23d474b82c021a7e9cf67eb-441pb"
      },
      "log": "fred_job: running job chicago-event id 1 run 11 ...\\n",
      "stream": "stdout"
    },
    "ingestionTime": "2021-06-18 07:19:17.610000 GMT+0000"
  },
  {
    "timestamp": "2021-06-18 07:19:15.085000 GMT+0000",
    "message": {
      "docker": {
        . . .
      }
    }
  }
]
```

See Also

fredcli batch get-results

fredcli batch get-results

Retrieve the results created for a given batch.

Synopsis

fredcli batch get-results

Downloads the results associated with *batch_id* from S3.

Previously this command accepted a *job_key* rather than *batch_id* as argument. Passing job keys to retrieve complete result sets will continue to work for backwards compatibility, but is considered deprecated. Retrieval of a single file with the *-file* option only works with a *batch_id*.

```
fredcli batch get-results <options> BATCH_ID
```

Options

--file <file>

Path to a single file to download from the results for *batch_id*, relative to the root of the results directory. E.g. *JOB/1/OUT/RUN1/CSV/infections.csv*.

Arguments

BATCH_ID

Required argument

Description

This command downloads the results generated for a batch. The results directory for the task associated with the given batch is downloaded to the current working directory, under a directory matching the batch id.

By default, the entire results directory is downloaded. If the *--file* option is specified, then only the single file specified by the path given to this option is downloaded. The file given for this option must be relative to the results directory for the given batch.

Examples

Here is an example of downloading the full results for a batch. In this example, the entire results directory would be stored in a created subdirectory eef4b59153ac480084a7ece647d855a9.

```
$ fredcli batch get-results eef4b59153ac480084a7ece647d855a9
Results unzipped and available as: eef4b59153ac480084a7ece647d855a9
```

As another example, the following command would retrieve just the total exposed results for this same batch. Note how the file path starts with the JOB directory. This result is available in the path eef4b59153ac480084a7ece647d855a9/results/JOB/1/OUT/PLOT/DAILY/INFLUENZA.totExposed.csv.

```
$ fredcli batch get-results eef4b59153ac480084a7ece647d855a9 \
> --file JOB/1/OUT/PLOT/DAILY/INFLUENZA.totExposed.csv
Results unzipped and available as: eef4b59153ac480084a7ece647d855a9
```

See Also

fredcli batch get-logs

fredcli batch get-status

Retrieve the status of a given batch.

Synopsis

fredcli batch get-status

Gets the current status of a batch.

```
fredcli batch get-status [options] BATCH_ID
```

Options

- output** <output>
The output format for this function.
 - Default** text
 - Options** text | json | yaml
- v, --verbose**
Show the detail view of the batch status.
 - Default** False

Arguments

BATCH_ID

Required argument

Description

Display the status for the given batch. This shows when the batch was submitted and when the current status was assigned. The `--output` option specifies the format for this command: one of `text`, `json`, or `yaml`.

If `-v` or `--verbose` is provided, then additional status information is shown.

Examples

The following command retrieves the status of an in-progress batch.

```
$ fredcli batch get-status 7bd3ede8a23d474b82c021a7e9cf67eb
Labels:
Submitted by: fred
Submitted on: 2021-07-07 17:59:50.652000 GMT+0000

Status set on: 2021-07-07 17:59:50.652000 GMT+0000
Status: Submitted

Tasks:
| ID | Status | Status set on |
|----|-----|-----|
| 65 | Submitted | 2021-07-07 17:59:50.652000 GMT+0000 |
```

As another example, the same status is shown with `yaml` output and `verbose` mode.

```
$ fredcli batch get-status 7bd3ede8a23d474b82c021a7e9cf67eb --verbose --output yaml
created: '2021-07-07T17:59:50.652Z'
createdBy: fred
guid: 5dfb38f9e4014490b8f9f718a58e23fc
labels: []
logEntry: ''
logTimestamp: 2021-07-07 17:59:50.652000 GMT+0000
status: Submitted
statusCreated: '2021-07-07T17:59:50.652Z'
submittedBy: fred
submittedOn: 2021-07-07 17:59:50.652000 GMT+0000
tasks:
- id: 65
  status: Submitted
  statusCreated: '2021-07-07T17:59:50.652Z'
```

See Also

fredcli batch get-logs

fredcli batch get-status-by-guid

This command is identical to `fredcli batch get-status`. See this command for details.

See Also

fredcli batch get-status

fredcli batch list

Display data about the most recent 20 batches run on the FRED platform.

Synopsis

fredcli batch list

Display data about the most recent 20 batches run on the FRED platform.

Returned data includes the batch's ID (guid), status, created time, and the email address of the submitting user.

```
fredcli batch list [options]
```

Options

--output <output>

Choose output format, either human readable *text*, or *json* or *yaml*.

Default text

Options text | json | yaml

--start <start>

The starting row to display from the list of all available batches.

Default 1

--count <count>

The number of rows to display.

Default 20

--user <user>

Choose a user to list batches for, if used without an argument it defaults to the user who is initiating the list batches request

Description

Returned data includes the batch's ID (guid), status, created time, and the email address of the submitting user. The `--output` option specifies the format for this command: one of `text`, `json`, or `yaml`. The `--user` option filters the batches queried by the argument given to `--user`. If no argument is given to `--user` it operates as a flag, and defaults to querying batches by the username associated with the request. The `--start` option when used in conjunction with the `--count` option will return only the dataset in the range of start to count. The `--start` and `--count` options default to 1 and 20 respectively. When this command is called with no arguments passed to `--start` or `--count`, the 20 most recent batches will be displayed.

Examples

The following command retrieves metadata about the most recently submitted batches in a tabular format. This is the `text` output format, and is the default.

```
$ fredcli batch list
|                               Batch |      Status |                Created |
↪Created By |   # Tasks |
|-----|-----|-----|-----|
↪-----|-----|
| b911d40bb95a44f99d1dccf072f8cdb7 | Succeeded | 28/07/2021, 06:26:07 PM | user.
↪1@epistemix.com |           1 |
| ce4d9c03badf48fda2218f1c5363ef1a | Succeeded | 23/07/2021, 05:09:43 PM | user.
↪2@epistemix.com |           1 |
| c5544406ddef4453903f32eea3f04cd6 | Succeeded | 23/07/2021, 04:50:46 PM | user.
↪1@epistemix.com |           1 |
...

```

As another example, the same batch list status is shown with `yaml` output

```
$ fredcli batch list --output yaml
- batchGuid: b911d40bb95a44f99d1dccf072f8cdb7
  created: '2021-07-28T18:26:07.480Z'
  createdBy: user.1@epistemix.com
  status: Succeeded
  tasks: 1
- batchGuid: ce4d9c03badf48fda2218f1c5363ef1a
  created: '2021-07-23T17:09:43.608Z'
  createdBy: user.2@epistemix.com
  status: Succeeded
  tasks: 1
- batchGuid: c5544406ddef4453903f32eea3f04cd6
  created: '2021-07-23T16:50:46.378Z'
  createdBy: user.1@epistemix.com
  status: Succeeded
  tasks: 1

```

In this example, the same batch list status is shown filtered by user `user.1@epistemix.com` with the default `text` output

```
$ fredcli batch list --user user.1@epistemix.com
|                               Batch |      Status |                Created |
↪Created By |   # Tasks |
|-----|-----|-----|-----|
↪-----|-----|

```

(continues on next page)

(continued from previous page)

```
| b911d40bb95a44f99d1dccb072f8cdb7 | Succeeded | 28/07/2021, 06:26:07 PM | user.  
↪1@epistemix.com | 1 |  
| c5544406ddef4453903f32eea3f04cd6 | Succeeded | 23/07/2021, 04:50:46 PM | user.  
↪1@epistemix.com | 1 |
```

Here, the same batch list status is shown with `--user` as a flag, with the default text output. This example assumes that the user initiating the request is `user.2@epistemix.com`.

```
$ fredcli batch list --user  
| Batch | Status | Created |  
↪Created By | # Tasks |  
|-----|-----|-----|  
↪-----|-----|  
| ce4d9c03badf48fda2218f1c5363ef1a | Succeeded | 23/07/2021, 05:09:43 PM | user.  
↪2@epistemix.com | 1 |
```

Here, the batch list is shown with `--start` and `--count` options set to 1 and 2 respectively, without filtering by a user.

```
$ fredcli batch list --start 1 --count 2  
| Batch | Status | Created |  
↪Created By | # Tasks |  
|-----|-----|-----|  
↪-----|-----|  
| b911d40bb95a44f99d1dccb072f8cdb7 | Succeeded | 28/07/2021, 06:26:07 PM | user.  
↪1@epistemix.com | 1 |  
| ce4d9c03badf48fda2218f1c5363ef1a | Succeeded | 23/07/2021, 05:09:43 PM | user.  
↪2@epistemix.com | 1 |
```

See Also

fredcli env get-status, fredcli batch get-status

fredcli batch submit

Submit the current directory as a FRED model.

Synopsis

fredcli batch submit

Execute the model in the current directory in FRED Cloud

```
fredcli batch submit [options]
```

Options

- compute-size** <compute_size>
[Deprecated] The compute size of the EKS container to execute simulation. [default: default]
- Options** tiny | small | default | medium | large | massive | small2 | default2 | medium2 | large2 | xlarge2 | small-mem | default-mem | medium-mem | large-mem | massive-mem
- s, --size** <size>
The compute size of the EKS container to execute simulation. [default: default]
- Options** tiny | small | default | medium | large | massive | small2 | default2 | medium2 | large2 | xlarge2 | small-mem | default-mem | medium-mem | large-mem | massive-mem
- config-file**
Tell fredcli to expect config file.
- queue**
Tell fredcli to queue batch instead of submitting directly to k8s.
- f, --file** <file>
Designate the config file to use.
- Default** fredconfig.json
- container-log-level** <container_log_level>
The log level to be used in the container running the simulation.
- Default** error
- Options** debug | info | warning | error | critical
- l, --label** <label>
Label the batch. Can have multiple labels by using this option multiple times. i.e. -label Label1
-label Label2
- k, --kill** <kill>
Amount of time to wait before killing the batch.Can be hours(h) or days(d). i.e. -k 24h is 24 hours,
-k 2d is two days.
- Default** 1d
- output** <output>
The output format for this function.
- Default** text
- Options** text | json | yaml
- help-size**
Show details on the available size options

Description

This command submits the FRED model in the current working directory to FRED Cloud for execution. By default, the command will look for an execute script called `fred-cloud-execute`. It is an error if this script is not found.

The entire directory is packaged and uploaded to FRED Cloud, except for ignored files (see below), and the execute script is called within a Docker container running FRED. For information on how to create an execute script, see the section *Batch Execution Scripts*.

If the `--config-file` option is specified, then the command looks for a configuration file instead of an execute script. In this case, the file `fredconfig.json` is used by default. The `-f` or `--file` option can be used to override this default and specify a name for the configuration file.

The size of the container is specified by the `--size` option. The list of possible container sizes are shown in the following tables. The specific sizes available to you can be displayed on the command-line with the option `--help-size`. This command also displays cost information based on your organization. The specific containers available and cost per hour vary depending on your FRED Cloud configuration.

Table 1: Standard Sizes

name	aws ami	cores	ram (GB)
tiny	r6gd.large	2	16
small	r6gd.xlarge	4	32
default	r6gd.2xlarge	8	64
medium	r6gd.4xlarge	16	128
large	r6gd.8xlarge	32	256
massive	r6gd.16xlarge	64	512

Table 2: Double Sizes

name	aws ami	cores	ram (GB)
small2	x2gd.xlarge	4	64
default2	r5ad.2xlarge	8	128
medium2	x2gd.4xlarge	16	256
large2	x2gd.8xlarge	32	512
xlarge2	x2gd.16xlarge	64	1024

Table 3: High Memory Sizes

name	aws ami	cores	ram (GB)
small-mem	x1e.xlarge	4	122
default-mem	x1e.2xlarge	8	244
medium-mem	x1e.4xlarge	16	488
large-mem	x1e.16xlarge	64	1952
massive-mem	x1e.32xlarge	128	3904

You can alter the amount of log information produced within FRED Cloud using the `--container-log-level` option. This affects the amount of information available from the `fredcli batch get-logs` command. The possible log level values are as follows.

- `critical` includes only critical errors in the logs
- `error` includes any error in the logs, and is the default
- `warning` includes both warning and error messages in the logs
- `info` includes informational messages as well as warning and errors in the logs

- `debug` includes detailed information about the processing of the batch in the logs, including information, warning, and error messages

You can label batches at submission by using the `--label` option. A batch can be assigned multiple labels by using the option repeatedly. For example if you would like to label a batch as “calibration” and “new-york” the command would be: `fredcli batch submit --label calibration --label new-york`. Labels can only contain alphanumeric characters, dashes(-) and/or underscores(_).

By default, FRED Cloud will wait up to 1 day (24 hours) for a batch to complete. After this day after submission has passed, the batch will be killed even if the batch only started processing recently. To extend (or reduce) the amount of time that FRED Cloud will wait for a batch to complete, use the `-k` or `--kill` option. This accepts a numeric value with a suffix of `h` for hours or `d` for days.

To ignore files or directories, a `.fredcli-ignore` file can be created in the model directory. Any files or directories listed in this file, using one item per line, will be ignored and not uploaded to FRED Cloud. By default, the `.git` directory and the `.fredcli-ignore` file are ignored, if they exist. Note that ignored items (files or directories) must be the actual name of the item to ignore; wildcards and other constructs are not supported in the ignore file at this time.

The final option available for this command is the `--output` option, with possible values `text`, `json`, or `yaml`. By default, the output option of `text` is used.

Examples

The following command runs the model in the current directory using the local `fred-cloud-execute` script in a `tiny` container in FRED Cloud.

```
$ fredcli batch submit -s tiny
Packaging model code.
Uploading model code.
Model code successfully uploaded.
Starting execution.
Your batch guid is cafdde0ea5bb4550be2242e7878032ea
To track the status of your batch use fredcli batch get-status.
↪cafdde0ea5bb4550be2242e7878032ea
```

Here is another example where the default configuration file `fredconfig.json` is used to execute the model with `json` output

```
$ fredcli batch submit --output json --config-file
{
  "batch_id": "843fc3e4dca64e9897944788a60733c3"
}
```

See Also

fredcli batch get-status, fredcli batch get-logs

fredcli batch validate-config

Verify that a batch configuration file is valid.

Synopsis

fredcli batch validate-config

Validate the structure of a batch configuration file.

```
fredcli batch validate-config [options]
```

Options

-f, --file <file>

The configuration file to check.

Default fredconfig.json

Description

This command validates a configuration file for use with the *fredcli batch submit* command. If the file is valid, a message is printed to the console and the command exits with a success code (0). If the file is not valid, the list of errors is printed and the command exits with an error code (non-zero, typically 1).

By default, the command looks for and evaluates the file `fredconfig.json` in the current working directory. The `-f / --file` option can be used to specify an alternate file to evaluate.

For a full discussion of configuration files, including the format requirements, please see the section *Batch Configuration Files* in the Guide.

Examples

In this example, a valid configuration file `fredconfig.json` (the default) is provided to the command.

```
$ fredcli batch validate-config
Configuration is valid.
```

Here is an example that invokes the command on the file `model-config.json` with an invalid configuration.

```
$ fredcli batch validate-config -file model-config.json
Configuration is invalid.
population.locations: missing property
locations: must be an array
job.startRun: must be a positive integer
```

See Also

fredcli batch submit, Batch Configuration Files

2.3 fredcli env

The `fredcli env` commands allow a modeler to interact with the FRED Cloud environment.

2.3.1 Synopsis

fredcli env

These are the environment functions for `fredcli`.

```
fredcli env [options] <subcommand>
```

Commands

get-status

Get usage and quota information for the...

2.3.2 Description

Environment commands relate to the overall status of the FRED Cloud environment. See the available commands for details.

2.3.3 Available Commands

fredcli env get-status

Retrieve status information for the FRED Cloud environment

Synopsis

fredcli env get-status

Get usage and quota information for the FRED cloud platform.

```
fredcli env get-status [options]
```

Options

--output <output>

Choose output format, either human readable *text*, or *json*.

Default text

Options text | json

Description

Display the nodes running and queued nodes in the FRED Cloud environment. This shows both standard and memory nodes, corresponding to the standard and high memory collection of sizes available in the *fredcli batch submit* command.

The following information is provided:

- Running Nodes is the number of EC2 nodes currently executing in the environment.
- Running CPUs is the total number of CPU cores across the running nodes. This also shows the maximum CPU quota for the indicated node type.
- Queued Batches is the batches queued and not yet running in the environment.
- Queued CPUs is the total number of CPU cores across the queued batches.

Note that a node in the process of starting up or shutting down may appear in both the Running and Queued categories. As such, the total number of CPUs may exceed the indicated maximum number of CPUs in such cases.

Examples

The following example shows the text output when nothing is running in the environment.

```
$ fredcli env get-status
|   TYPE   |  RUNNING NODES  |  RUNNING CPUS  |  QUEUED BATCHES  |  QUEUED CPUS  |
|-----|-----|-----|-----|-----|
| standard |           0     |    0 / 3536   |           0     |           0     |
| memory   |           0     |    0 / 512    |           0     |           0     |
$
```

As another example, here the same status with output type JSON.

```
$ fredcli env get-status --output json
{"quotas": {"standard": {"cpu": 3536}, "memory": {"cpu": 512}}, "runningNodes":
{"standard": {"count": 0, "cpu": 0}, "memory": {"count": 0, "cpu": 0}}, "queued
Batches": {"standard": {"count": 0, "cpu": 0}, "memory": {"count": 0, "cpu": 0}
}}
$
```

See Also

fredcli batch get-status

2.4 fredcli user

The *fredcli user* commands allow a modeler to interact with their user information, including the ability to log in to the FRED Cloud environment.

2.4.1 Synopsis

fredcli user

These are the user commands

```
fredcli user [options] <subcommand>
```

Commands

login

Stores credentials for using fredcli

2.4.2 Description

The *fredcli user* commands perform actions on the user's account. In particular, the user must log in to the FRED Cloud environment before other *fredcli* commands can be successfully performed.

See the available commands for details on the available actions.

2.4.3 Available Commands

fredcli user login

Logs a user into the FRED Cloud environment using an authorization token.

Synopsis

fredcli user login

Stores credentials for using fredcli

```
fredcli user login [options] AUTH_TOKEN
```

Options

--overwrite-existing

Flag to overwrite existing FREDCLI authorization token

Arguments

AUTH_TOKEN

Required argument

Description

In order to perform `fredcli` commands, a user must be authorized for the FRED Cloud environment. This is currently done using an authorization token generated from [FRED Studio](#).

Once an authorization token is allocated, the environment must be initialized for `fredcli` using this command. The authorization token is provided as the only argument for the command.

If successful, the authorization token is stored in a file `config.json` in the directory `.fredcli` of the user's home directory. When other `fredcli` commands are executed, the presence and validity of this authorization token is verified before the requested command is executed.

Errors

If a token is already assigned, an error is shown if the `--overwrite-existing` option is not provided.

Examples

The following example stores the authorization token in the environment.

```
$ fredcli user login
→ eyJhbGciOiJIUzI1NiIsInR5cCI6IiA6IiY2JjY2UyZi0xMTZkLTQ4NjQtOGNkMi1kYTQ2OGQxMzBjYTAif
→ eyJpYXQiOiJlZmJlYmJjc0MzUsImp0aSI6IjMxZTAzMTg3LTc3ZTUtNDRiZS1iYTRmLTk5MjMxMjFiMDQwYyIsIm
$
```

If a token was previously assigned, then the `--overwrite-existing` option should be used to indicate the token should be overwritten, as in the following.

```
$ fredcli user login
→ eyJhbGciOiJIUzI1NiIsInR5cCI6IiA6IiY2JjY2UyZi0xMTZkLTQ4NjQtOGNkMi1kYTQ2OGQxMzBjYTAif
→ eyJpYXQiOiJlZmJlYmJjc0MzUsImp0aSI6IjMxZTAzMTg3LTc3ZTUtNDRiZS1iYTRmLTk5MjMxMjFiMDQwYyIsIm
→ --overwrite-existing
2021-06-11 20:46:11,428 WARNING:Overwriting existing FREDCLI authorization token
$
```

Symbols

- compute-size <compute_size>
fredcli-batch-submit command line option, 15
- config-file
fredcli-batch-submit command line option, 15
- container-log-level <container_log_level>
fredcli-batch-submit command line option, 15
- count <count>
fredcli-batch-list command line option, 12
- file <file>
fredcli-batch-get-results command line option, 9
fredcli-batch-submit command line option, 15
fredcli-batch-validate-config command line option, 18
- help-size
fredcli-batch-submit command line option, 15
- interval <interval>
fredcli-batch-get-logs-command command line option, 7
- kill <kill>
fredcli-batch-submit command line option, 15
- label <label>
fredcli-batch-submit command line option, 15
- limit <limit>
fredcli-batch-get-logs-command command line option, 7
- output <output>
fredcli-batch-get-logs-command command line option, 7
fredcli-batch-get-status command line option, 10
fredcli-batch-list command line option, 12
- fredcli-batch-submit command line option, 15
- fredcli-env-get-status command line option, 20
- overwrite-existing
fredcli-user-login command line option, 22
- queue
fredcli-batch-submit command line option, 15
- size <size>
fredcli-batch-submit command line option, 15
- start <start>
fredcli-batch-list command line option, 12
- user <user>
fredcli-batch-list command line option, 12
- verbose
fredcli-batch-get-logs-command command line option, 7
fredcli-batch-get-status command line option, 10
- f
fredcli-batch-submit command line option, 15
fredcli-batch-validate-config command line option, 18
- i
fredcli-batch-get-logs-command command line option, 7
- k
fredcli-batch-submit command line option, 15
- l
fredcli-batch-get-logs-command command line option, 7
fredcli-batch-submit command line option, 15
- s
fredcli-batch-submit command line

option, 15

-v
fredcli-batch-get-logs-command command line option, 7
fredcli-batch-get-status command line option, 10

A

AUTH_TOKEN
fredcli-user-login command line option, 22

B

BATCH_ID
fredcli-batch-get-logs-command command line option, 7
fredcli-batch-get-results command line option, 9
fredcli-batch-get-status command line option, 11

F

fredcli-batch-get-logs-command command line option
--interval <interval>, 7
--limit <limit>, 7
--output <output>, 7
--verbose, 7
-i, 7
-l, 7
-v, 7
BATCH_ID, 7

fredcli-batch-get-results command line option
--file <file>, 9
BATCH_ID, 9

fredcli-batch-get-status command line option
--output <output>, 10
--verbose, 10
-v, 10
BATCH_ID, 11

fredcli-batch-list command line option
--count <count>, 12
--output <output>, 12
--start <start>, 12
--user <user>, 12

fredcli-batch-submit command line option
--compute-size <compute_size>, 15
--config-file, 15
--container-log-level <container_log_level>, 15
--file <file>, 15
--help-size, 15
--kill <kill>, 15
--label <label>, 15
--output <output>, 15
--queue, 15
--size <size>, 15
-f, 15
-k, 15
-l, 15
-s, 15
fredcli-batch-validate-config command line option
--file <file>, 18
-f, 18
fredcli-env-get-status command line option
--output <output>, 20
fredcli-user-login command line option
--overwrite-existing, 22
AUTH_TOKEN, 22