GISAID Uploader – Command Line Interface ("CLI")

User Information and Manual

Introduction:

To facilitate both workflow and submission processes for higher volumes of data, an automated, streamlined process is available via a Command Line Interface ("CLI").

The CLI may be tested with small submissions (<100 sequences), using a TEST-Client-ID (TEST-EA76875B00C3). Testing the CLI feature will not affect the live system; it will only test the functionality of the CLI with the system. One practice submission with the Test-Client-ID is required before requesting a personalized Client-ID.

The personalized Client-ID can be requested by emailing CLIsupport@gisaid.org

Prerequisites:

- Python 3.8 minimum (Python3.5+ supported with additional "requests" library installed)
- Linux/Windows/macOS operating system
- Internet connection to access the GISAID GPS-API

Get help:

```
./gisaid_uploader --help
```

Authenticate once:

```
./gisaid_uploader CoV authenticate --cid TEST-EA76875B00C3
```

Repeat daily or weekly:

```
./gisaid_uploader CoV upload --fasta sequences.fasta --csv
metadata.csv --failedout failed_metadata.csv
```

Detailed Description and Usage:

The "gisaid_uploader" python script facilitates submissions (CSV- and FASTA-files) to the EpiCoV™ curation-zone, employing a single python-script that uses the GISAID GPS-API to authenticate and upload data. It consists of one self-contained python script that only uses standard Python 3.8 libraries and can be downloaded here.

For the correct CSV-File-Format and FASTA-File-Format consult the GISAID EpiCoV™ website: (Go to: EpiCoV™ > Upload > Batch Upload > Download Instructions and Template)

Authentication is completed upfront via username/password and persists in an authentication-token, that is valid for 100 days.

The "gisaid_uploader" Python script can be used interactively from the command line. It also can output a JSON-formatted log-file and accept all parameters as command-line-arguments and so suitable be called from other programs. Experienced programmers may also use this script as a python module and pass all configuration parameters as an ArgumentParser-Style object.

All operations require a "context" to be given as a first positional parameter. Currently the only valid context is "CoV".

All data submitted using the TEST-Client-ID (**TEST-EA76875B00C3) are only sent to a Test-System and will not be released.**

Examples and Workflow

1. Authenticate

Create an authentication token and save it to a local file for uploading data:

```
./gisaid_uploader CoV authenticate -cid TEST-EA76875B00C3 --user your-gisaid-username
```

Interactively asks missing parameters (i.e., the user's password) and saves the authentication-token to a local file in the current directory (default name: gisaid_uploader.authtoken)

A GISAID username and password is required to test-upload with the TEST-Client-ID (TEST-EA76875B00C3). All submissions using the TEST-Client-ID will be sent to a test-system and will not be released in the GISAID EpiCoV-live system.

To upload data intended for release to the GISAID EpiCoV-live-system, a personalized Client-ID must be requested by emailing CLIsupport@gisaid.org

Output if "-l" not provided:

```
valid until e.g.: Sun Mar 28 12:31:26 2021
user authenticated
```

or

```
could not authenticate user
```

or

```
unknown client-id
```

2. Upload

If the authentication was successful and the access-token-file is available in the current directory, or the option (-a / --authfile) is given and points to the auth-file, or the option (-t / --authtoken) is given and contains the authtoken itself, use the action "upload" to upload sequences:

```
./gisaid_uploader CoV upload --fasta sequences.fasta --csv metadata.csv
```

Since the authtoken is valid for several months, you may perform daily or weekly uploads without reauthentication from the machine.

Output of this operation is a mapping of virus names to EPI_ISL IDs and validation errors per virus name. All virus submissions that return an EPI_ISL ID, are correctly uploaded into the GISAID EpiCoV curation space and must not be submitted again.

For convenience, the command-line argument (-f / --failedout) can be supplied with a file name. This file will contain the metadata (from the CSV-file) of the failed uploads. It is safe to use this file with the original FASTA-File - after correcting the meta-data-validation-errors - to upload again.

```
./gisaid_uploader CoV upload --fasta sequences.fasta --csv metadata.csv --
failedout failed_metadata.csv
```

Revoking authentication token (optional)

All authtokens can be revoked by simply supplying the client-id:

```
./gisaid_uploader CoV revoke --cid TEST-EA76875B00C3
```

Codes and Messages

Code	Message	
version		
valid_until	valid until: DATE	
ok	user_authenticated	For command “authenticate”
unknown_cid	unknown client-id	
no_auth	could not authenticate user	
error	unexpected response from GPS-API	
invalid_params	--authtoken not supplied or auth-file not found or supplied authtoken invalid for this context or no --cid supplied	
ok	all auth-tokens successfully revoked	For command “revoke”
ok	no auth-tokens found	For command “revoke”
unknown_cid	unknown client-id	
invalid_params	--authtoken not supplied or auth-file not found or supplied authtoken invalid for this context	
fasta_format_error		FASTA-File contains invalid data or format
missing_seq	missing sequence for VIRUS-NAME	In FASTA File
invalid_token	invalid auth-token	
epi_isl_id	VIRUS-NAME; EPI_ISL_ID	Message can be split on “;” to get the mapping between virus-name and EPI_ISL_ID
validation_error	VIRUS-NAME; ERROR-CODE; ERROR-DETAIL	Message can be split on “;” to get virus-name, error-code and error-detail (JSON-Object)
upload_error	VIRUS-NAME; ERROR-CODE	Message can be split on “;” to get virus-name, error-code
upload_count	submissions uploaded: COUNT	
failed_count	submissions failed: COUNT	