README.md

github_cli

github_cli is a Python(3) library and cli tool to access the GitHub API v3 and [Github Enterprise API v3]. This library enables you to query and rank GitHub resources such as an organization's repositories, by number of stars, number of forks, etc.

Setup

1. Prerequisites:
   - For running tests, we will need the pytest module to be installed(follow instructions on the 'Running Tests' section)
   - For fetching data from the web, we use the requests library
   - For caching expensive network calls, we use the requests_cache

   ```
   pip3 install pytest
   pip3 install requests
   pip3 install requests_cache
   ```

2. Copy over the zip file, or extracted relevant files to directory of choice. Unzip and enter root directory. Ensure normal read write permissions at the root directory level all the way down
3. Obtain a token by following instructions on: Procuring Access Token. Ensure that the token is generated with access to public repos
4. Copy over the token to the config.json file, under the key 'api_token'
5. For running tests, refer to section named 'Running Tests' in this document

Install

1. To install the tool, from the root directory, execute the following commands:

```
chmod 747 install.sh
./install.sh
```

Alternatively, you can directly use pip for install. From the root directory, execute the following command:

```
pip3 install -e .
```

To verify successful installation, make sure the root directory contains a directory named 'github_cli.egg-info/'. This ensures that the install has happened successfully.

2. To uninstall the tool, from the root directory, execute the following commands:

```
chmod 747 uninstall.sh
./uninstall.sh
```

Alternatively, you can directly use pip for install. From the root directory, execute the following command:

```
pip3 uninstall github_cli
```

3. To clean the environment, delete all temp files and caches, from the root directory, execute the following commands:

```
chmod 747 clean.sh
./clean.sh
```

Usage

After following instructions from the 'Setup' section, simply run the command line tool.

Start by making sure that the API_TOKEN is specified directly in the config.json file.

1. To get the top 10 repos by number of stars:

```
github_cli -org netflix -s 10
```

2. To get the top 5 repos by number of forks:

```
github_cli -org netflix -f 5
```

3. To get the top 5 repos by number of pull requests:

```
github_cli -org netflix -pr 5
```

4. To get the top 5 repos by contribution percentage:

```
github_cli -org netflix -cp 5
```

Running manually

Alternatively, if the install hasn't happened, and for a manual run, execute the following program from the root directory, exactly with the same parameters mentioned above:

```
python github_cli_tool.py -org netflix -s 10
python github_cli_tool.py -org netflix -f 10
python github_cli_tool.py -org netflix -pr 10
python github_cli_tool.py -org netflix -cp 10
```

To use as library

At the root directory, create your test program (or follow the examples in test_github_cli.py):

```python
from github_cli import GithubCli

# First create a GithubCli instance, using access token
# If no token is explicitly mentioned,
# we will assume the config.json has it
gcli = GithubCli("API_TOKEN")

# Obtain top 10 repos by number of stars
print(gcli.get_top_repos_by_stars(organization='netflix', number_of_repos=10))
# Obtain top 10 repos by number of forks
print(gcli.get_top_repos_by_forks(organization='netflix', number_of_repos=10))
# Obtain top 10 repos by number of pull requests (PR)
print(gcli.get_top_repos_by_pull_requests(organization='netflix', number_of_repos=10))
```

```
# Obtain top 10 repos by contribution percentage
print(gcli.get_top_repos_by_contribution_percentage(organization='netflix', number_of_repos=10))
```

Running Tests

For testing, we would require the pytest module. Can be installed fairly quickly via pip:

```
pip3 install pytest
```

Once this dependency has been installed, simply run pytest to run all unit tests.

```
pytest tests.py
```