TrelloZilla Documentation

Release 0.1.3

Moisés Guimarães de Medeiros

Jan 18, 2021

Contents:

TrelloZilla

TrelloZilla is a small set of boilerplate code to speed up automations in/between both Bugzilla and Trello.

- Free software: Apache Software License 2.0
- Documentation: https://trellozilla.readthedocs.io.

1.1 Features

- Configurable via oslo.config;
- Generates Trello access tokens;
- Creates Trello API objects based on provided config;
- Creates Bugzilla API objects based on provided config.

Installation

2.1 stable release

To install TrelloZilla, run this command in your terminal:

```
$ pip install trellozilla
```

This is the preferred method to install TrelloZilla, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 from sources

The sources for TrelloZilla can be downloaded from our git repo:

```
$ git clone git://github.com/moisesguimaraes/trellozilla
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

Usage

TrelloZilla requires you to provide configuration values for both Trello's and Bugzilla's API. The sample configuration file is well documented and self-explanatory about how you can produce the right values for it.

3.1 Generating configuration files

To generate sample configuration files:

```
$ mkdir -p ~/.trellozilla/trellozilla.conf.d

$ cd ~/.trellozilla/trellozilla.conf.d

$ oslo-config-generator --namespace trellozilla > trellozilla.conf

$ oslo-config-generator --namespace oslo.log > logging.conf
```

Check the oslo.config documentation to learn more about other places where you can place your configuration files.

3.2 The Trello configuration

The Trello configuration is composed of four values:

- api_key
- api_secret
- token
- token_secret

The first two ones you can find at https://trello.com/app-key.

Once you have setup your configuration file with your Trello api_key and api_secret you can use the following command to generate the token and token_secret:

```
$ generate_trello_oauth_token
```

3.3 The Bugzilla configuration

The Bugzilla configuration is composed of two values:

- url
- api_key

The buggzila api_key can be generated at:

your_bugzilla_url/userprefs.cgi?tab=apikey

3.4 Getting the API objects

To use TrelloZilla in a project:

```python
import trellozilla

conf = trellozilla.get_config()

conf("trellozilla")  # or `conf(project="my_project")` to help oslo.config finding
→your config file.

trello = trellozilla.trello_api(conf)
bugzilla = trellozilla.bugzilla_api(conf)
```

For following usage of both Trello and Bugzilla API objects, please refeer to py-trello and python-bugzilla projects.

CHAPTER 4

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/moisesguimaraes/trellozilla/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

4.1.4 Write Documentation

TrelloZilla could always use more documentation, whether as part of the official TrelloZilla docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/moisesguimaraes/trellozilla/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *trellozilla* for local development.

1. Fork the *trellozilla* repo on GitHub.

2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/trellozilla.git
   ```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv trellozilla
$ cd trellozilla/
$ pip install -e .
```

4. Create a branch for local development:

   ```
   $ git checkout -b name-of-your-bugfix-or-feature
   ```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests with supported Python versions using tox:

   ```
   $ tox
   ```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

   ```
   $ git add .
   $ git commit -m "Your detailed description of your changes."
   $ git push origin name-of-your-bugfix-or-feature
   ```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.

2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.

3. The pull request should work for Python 3.6+. Check https://travis-ci.org/moisesguimaraes/trellozilla/pull_requests and make sure that the tests pass for all supported Python versions.

CHAPTER 5

Authors

Indices and tables

- genindex
- modindex

Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.