cheddar Documentation

Release 1.0

Jesse Myers

November 19, 2013

Contents

Contents:

CHAPTER 1

cheddar

PyPI clone with Flask and Redis. It's the single most popular cheese in the world!

- Free software: Apache License V2
- Documentation: http://cheddar.rtfd.org.

1.1 Features

Cheddar aims to simplify Python development within organizations that simultaneously work with public and private Python distributions.

Cheddar includes:

- A *local* package index for internal development, supporting setuptools `register` and `upload` commands.
- A *remote* package index that proxies to a public repository (such as `pypi.python.org`) and *caches* packages and package version listings to reduce latency and minimize the effect of downtime by the public repository.
- A *combined* package index that unifies the best of the local and remote implementations.

In addition, Cheddar supports a few features that simplify management within an organization:

- Packages are stored locally in separate directories for pre-releases and releases, simplifying backup strategies that wish to ignore transitive development builds.
- Duplicate package uploads return a predictable HTTP *409 Conflict* error.
- Mistakenly uploaded packages may be deleted using a simple, RESTful API.

1.2 Configuration

Cheddar can run in any WSGI container or through Flask's built-in development server (which is single-threaded and only recommended for development).

Configuation is loaded from the `defaults.py` module along with the contents of the file pointed to by the `CHEDDAR_SETTINGS` environment variable, if any.

You may wish to modify several of the configuration parameters from their default values, including:

- *INDEX_URL* which specifies the URL of the *remote* package index
- *REDIS_HOSTNAME* which control the location of the Redis server
- *LOCAL_CACHE_DIR* which controls the storage location of locally uploaded files
- *REMOTE_CACHE_DIR* which controls the storage location of cached remote files

1.3 The Local Index

To use the local index:

1. Edit your `~/.pypirc` to contain an entry for Cheddar. It should look _something_ like:

   ```
   [distutils]
   index-servers =
       pypi
       cheddar

   [pypi]
   repository:http://pypi.python.org

   [cheddar]
   repository:http://localhost:5000/pypi
   username:myusername
   password:mypassword
   ```

   Note that the URL here assume you are running the "development" server.

2. Add credentials to Redis:

   ```
   redis-cli set cheddar.user.myusername mypasswod
   ```

3. Upload your distribution:

   ```
   cd /path/to/directory/containing/setup.py
   python setup.py sdist upload -r cheddar
   ```

   You may also use the `register -r cheddar` to validate your `setup.py` without uploading the source distribution.

1.4 The Remote Index

Run *pip* using a custom index url:

```
pip install --index-url http://localhost:5000/simple
```

Note that the URL here assume you are running the "development" server.

You can also edit your `~/.pip/pip.conf` to contain the index url automatically:

```
[install]
index-url = http://localhost:5000/simple
```

1.5 Data

Cheddar saves data in several places:

- Local packages are stored in the *LOCAL_CACHE_DIR*
- Remote packages may be cached in the *REMOTE_CACHE_DIR*
- Remote version listings may be cached in Redis.
- User data (for upload authentication) is stored in Redis.
- Local package version listings are stored Redis.

CHAPTER 2

Installation

At the command line:

```
$ pip install cheddar
```

CHAPTER 3

Usage

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/jessemyers/cheddar/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" is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

4.1.4 Write Documentation

cheddar could always use more documentation, whether as part of the official cheddar 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/jessemyers/cheddar/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 *cheddar* for local development.

1. Fork the *cheddar* repo on GitHub.
2. Clone your fork locally:

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

3. Install your local copy into a virtualenv. this is how you set up your fork for local development:

   ```
   $ virtualenv venv
   $ source venv/bin/activate
   $ python setup.py develop
   ```

4. Create a branch for local development:

   ```
   $ git flow feature start 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, including testing other Python versions with tox:

```
  $ flake8 cheddar tests
        $ python setup.py test
  $ 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 flow feature finish 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:

Cheddar uses gitflow for its branch management.

1. Implement changes in new git branches, following git-flow's model:

- Changes based off of *develop* will receive the least amount of skepticism.
- Changes based off of a *release* branches (if one exists) will be considered, especially for small bug fixes relevant to the release. We are not likely to accept new features against *release* branches.
- Changes based off of *master* or a prior release tag will be given the most skepticism. We may accept patches for major bugs against past releases, but would prefer to see such changes follow the normal git-flow process.

    We will not accept new features based off of *master*.

2. Limit the scope of changes to a single bug fix or feature per branch.
3. Treat documentation and unit tests as an essential part of any change.
4. Update the change log appropriately.
5. The pull request should work for Python 2.7 and PyPy Check https://travis-ci.org/jessemyers/cheddar/pull_requests and make sure that the tests pass for all supported Python versions.

CHAPTER 5

Credits

5.1 Development Lead

- Jesse Myers https://github.com/jessemyers

5.2 Contributors

- Luis Morales https://github.com/lacion

CHAPTER 6

History

6.1 1.0 (2013-11-19)

- Initial version.

CHAPTER 7

Indices and tables

- *genindex*
- *modindex*
- *search*