
Python Issue Tracking

Release 0.1

Sep 11, 2019

Contents

1	Issues	1
1.1	Current - bugs.python.org	1
1.2	PEPs	1
1.3	Roadmap - BPO Improvement	1
1.4	Migration - GitHub Issue Integration/Management	1
2	Repos of bpo	3
2.1	Roundup - Upstream Code Base	3
2.2	Roundup Python - modifications and extensions	3
2.3	bpo instance - used for deployment	4
2.4	Development Bootstrap with Docker	4
3	SRE Guide	5
3.1	Deployment details	5
4	Dev Environment	7
4.1	Bootstrap using Docker	7
4.2	From scratch	8
5	London Core Sprint	9
5.1	Issues Brainstorming and Resources	9
5.2	Scripts and Proof of concepts	9
5.3	Priorities	9
5.4	Challenges with Roundup	10
6	Indices and tables	13

1.1 Current - bugs.python.org

<https://bugs.python.org>

Links on the bottom left of the page provide help information.

1.2 PEPs

PEP 581 - Using GitHub Issues for CPython **Accepted**

PEP 588 - GitHub migration plan

PEP 595 - Improving bugs.python.org

1.3 Roadmap - BPO Improvement

1.4 Migration - GitHub Issue Integration/Management

CHAPTER 2

Repos of bpo

This document attempts to capture the repos associated with bugs.python.org. It is ordered in a rough progression from vanilla upstream to customized bpo.

2.1 Roundup - Upstream Code Base

This is the base Roundup project.

Official Sourceforge: <https://sourceforge.net/p/roundup/code/ci/default/tree/>

- Current release on PyPI: 1.6.0
- Head of master (2.0dev) includes REST, Python 3 support

Mirror on GitHub: <https://github.com/roundup-tracker/roundup>

Mirror is kept up-to-date.

2.2 Roundup Python - modifications and extensions

Bitbucket Project has several repos containing roundup-python-fork, rietveld, and instances Project Organization: <https://bitbucket.org/account/user/python/projects/BPO>

2.2.1 Roundup - Python fork

Official Python Roundup fork with modifications

- Repo: <https://bitbucket.org/account/user/python/projects/BPO/roundup>

This fork of upstream roundup includes the base Roundup project with additional extensions for GitHub integration to PRs, etc.

- Roundup upstream bugs-python-org branch <https://hg.python.org/tracker/roundup/shortlog/bugs.python.org>

The diff between the Python fork and Upstream is now relatively small.

2.2.2 Rietveld - review tool

<https://bitbucket.org/account/user/python/projects/BPO/rietveld>

2.3 bpo instance - used for deployment

<https://bitbucket.org/account/user/python/projects/BPO/tracker/cpython>

2.3.1 Additional instances hosted for others

- Jython <https://bitbucket.org/account/user/python/projects/BPO/tracker/jython>
 - hosting of roundup's website and tracker <https://bitbucket.org/account/user/python/projects/BPO/tracker/roundup>
-

2.4 Development Bootstrap with Docker

Docker image docker-bpo: <https://github.com/python/docker-bpo>

3.1 Deployment details

CHAPTER 4

Dev Environment

This section describes setting up a dev environment.

4.1 Bootstap using Docker

Docker image docker-bpo: <https://github.com/python/docker-bpo>

Docker Hub image: <https://hub.docker.com/r/python/docker-bpo>

Another docker image (not sure its purpose - openshift deploy?) <https://hub.docker.com/r/python/bpo-builder>

4.1.1 Quickstart

```
git clone https://github.com/python/docker-bpo.git
cd docker-bpo
docker pull docker.io/python/docker-bpo
./pull.sh
# docker run [OPTIONS] IMAGE[:TAG|@DIGEST] [COMMAND] [ARG...]
# --rm: clean up container on exit
# -it: interactive tty
# -p: publish container port
# -v: volume to mount
# IMAGE: docker.io/python/docker-bpo
docker run --rm -it -p 9999:9999 -v `pwd`: /opt/tracker docker.io/python/docker-bpo

# After container mounts the volume
rd-start
```

Three accounts are created: user, developer, coordinator. Login with pass as password.

To view the user interface: <http://localhost:9999>

4.2 From scratch

Python wiki for Tracker - out of date: <https://wiki.python.org/moin/TrackerDevelopment>

4.2.1 Set Up by Components

Running tests from the repo::

```
python2 run_tests.py
```

4.2.2 Customizing Roundup

<http://roundup.sourceforge.net/docs/customizing.html>

5.1 Issues Brainstorming and Resources

5.2 Scripts and Proof of concepts

- Search UI using JSON
- Search personal issues
- XML RPC to Jupyter script
- [XMLRPC docs](#)

5.3 Priorities

1. Update Roundup
 - Update our clone from upstream, solve conflicts
 - [Update the instances](#)
 - Test and push the changes on bpo
2. GitHub Login
 - Create new tokens (ask Ernest)
 - Review/test/merge PR
 - Push the changes on bpo
 - (Contribute it upstream)
3. Rest API
 - Enable and test it after updating Roundup

4. UI update
 - Review Roundup responsive theme, GSoC student work, [wiki discussions](#) (see also [another wiki page](#) and the pages linked at the top)
 - Material design CSS <https://material.io/>
 - Mobile friendly theme
 5. Write REST tools
 6. Move bpo from bitbucket/mercurial to github/git
-

5.4 Challenges with Roundup

5.4.1 Current Deployment is brittle

Upstream Roundup

Sourceforge Repo

To clone with Mercurial (read-only): `hg clone http://hg.code.sf.net/p/roundup/code`

An up-to-date [mirror on GitHub](#) exists

Pros:

- Adding REST API
- Adding Python 3

Cons:

- Maintainers are few
- Complex old PHP UI frontend
- How ready is it for Python 3?

Decision points:

- which base: roundup, GitHub issues, or other (Bugzilla?)
- pay down technical debt now versus future (moving from round up later will likely be an even bigger undertaking)
- CRITICAL: Confidence in solid Python 3 support

Customized Python Roundup for bpo

Bitbucket Project Repos containing roundup*, rietveld, etc

*Clone of <http://roundup.sourceforge.net/> for bugs.python.org

To clone with Mercurial: `hg clone https://bitbucket.org/python/roundup`

Pros:

- Familiar workflow for long time contributors
- Better email support than GitHub for personal issue workflows

- Existing proof of concepts that we could add for usability as standalone web pages to the existing bpo navigation
 - Search UI using JSON
 - Search personal issues
- Addition of GitHub OAuth for user login

Cons:

- Dated workflow and tools discourage new contributors especially from underrepresented groups
- Unfriendly and unproductive UI
- Modernizing UI
 - requires rewrite of html/css/sass with material design
 - ripping out all of the old PHP Template Attribute Language code which results in brittle, barely maintainable UI (CRITICAL)
 - PyPI warehouse css would be an excellent foundation to work from
- Upstream changes to roundup could break any improvements (best option: permanent fork of roundup with goal to rewrite in Python 3 and Django/Pyramid)

Action item:

- Migrating to GitHub and git from mercurial would increase the number of contributors and be more appealing to front end developers in our community. **Do at the sprint this week?**

docker-bpo

Bootstrap a development environment using Docker

Repo: <https://github.com/python/docker-bpo>

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`