

---

# PSF: Integrate and test Roundup's REST API on bugs.python.org

## About me

**Enrico Carbognani** (bitbucket and github nickname **naccht**)

University of Luxembourg / Bachelor in Computer Science (BiCS)/ first year/ graduation in 2021

Time zone: Central European Summer Time (CEST)

<https://bitbucket.org/naccht/projects/>

## Code contribution

Here is the link to the application (Phenology Calendar App) I have developed for my first Bachelor Semester Project (BSP):

<https://bitbucket.org/naccht/phenologycalendarapp/src/default/>

Other code can be found at:

<https://bitbucket.org/naccht/projects/>

Pull request made to python-blogs:

<https://github.com/python-gsoc/python-blogs/pull/109>

Pull request made to cpython:

<https://github.com/python/cpython/pull/12866>

## Project information

---

**Sub-org name: PSF**

### **Project Abstract**

Integrate the REST API that is being added to upstream Roundup to bugs.python.org (b.p.o). Develop the following list of tools and features that employ the REST API:

- Tools to easily generate random issues/messages, useful for testing
- Command-line interface to access the tracker
- Advanced statistics, graphs, leaderboards
- Real time filtering/ordering of issues
- Customizable dashboard
- Related issues suggestions
- Simpler integrations with other services (GitHub, IRC, etc.)

### **Detailed description**

As reported here:

<https://mail.python.org/pipermail/core-workflow/2015-September/000220.html>

The REST API has been developed as part of a PSF GSoC 2015 project.

And, as reported here:

<https://issues.roundup-tracker.org/issue2550734>

The GSoC code was merged to the mainline of Roundup in:

<https://sourceforge.net/p/roundup/code/ci/75cfee324d006ae6c7406693c432f077164dcd52/>

and it is part of any Roundup post 1.6.0 release. The main activity for the REST API integration is updating b.p.o. to a post 1.6.0 release of Roundup.

---

After the porting it will be checked that none of the existing test are broken and fixes will be made as needed.

Then the new required tools and features will be implemented one by one and the existing REST API will be checked against. In case it will be fixed or improved and ported upstream in Roundup.

## **Weekly timeline**

### Community Bonding (May 7-26):

Self study, familiarize myself with Roundup. Get an understanding of the specific bugs.python.org (b.p.o) instance. Experimentation and testing. During this period, I will remain active on IRC and in touch with the mentor to discuss and finalize on how to proceed with the project. Start the definition of a Software Requirements Specification document which will include Use-Cases to be used as a reference for the development of the unit/regression tests.

### Week 1 (May 27-31):

Finalize Software Requirements Specification document. Follow the documentation here: <https://wiki.python.org/moin/TrackerDevelopment> and complete my own set up of b.p.o as test tracker where all changes will be developed and tested.

### Week 2 (June 3):

Update my installation of test tracker to the last (1.6.0) version of Roundup

### Week 3 (June 10):

Make sure the existing regression test suite pass completely and make fixes if needed

### Week 4 (June 17):

---

Develop the capability to easily generate random issues/messages.  
Correspondingly improve and fix the REST API as needed

Week 5 (June 24):

Develop the command-line interface to access the tracker. Correspondingly improve and fix the REST API as needed

Week 6 (July 1):

Add Advanced statistics, graphs and leaderboards. Correspondingly improve and fix the REST API as needed

Week 7 (July 8):

Adding Real time filtering/ordering of issues and Customizable dashboard.  
Correspondingly improve and fix the REST API as needed

Week 8 (July 15):

Provide capability to make suggestions on Related issues. Correspondingly improve and fix the REST API as needed.

Start the implementation of tests cases covering the new functionalities added.

Week 9 (July 22):

Provide a simpler integrations with other services (GitHub, IRC, etc.).  
Correspondingly improve and fix the REST API as needed.

Continue implementation of tests cases covering the new functionalities added.

Week 10 (July 29):

Finalize tests cases for full coverage. Start documentation of newly added functionalities.

---

Deploy changes made on test tracker to b.p.o. and check for any possible remaining issue.

Week 11 (August 5):

Code review for compliance with programming best practices. Write a few remaining test cases and test them. Proceed with "code freeze" and complete any tests on b.p.o. Proceed with documentation.

Week 12 (August 12):

Improve the documentation.

Final week (August 19):

Project submission

**Other commitments**

Follow courses and study for the final exams. Courses end on May 31. Exams run from June 1st to July 6th.

**Additional application**

I have wrote a proposal also to the Libre Space Foundation.