Scrumbugz Documentation

Release git master

Mozilla Foundation and contributors

April 28, 2015

Contents

CHAPTER 1

About Scrumbugz

This is the software that runs https://scrumbu.gs/.

This allows you to manage sprints backed by Bugzilla bug data.

CHAPTER 2

Site Administrator's Guide

2.1 Installing Scrumbugz

TODO: Write this.

For now, follow the *Hacking HOWTO*.

CHAPTER 3

Contributor's Guide

3.1 Join this project!

Interested in building a scrummy Bugzilla? Interested in running one? Then you should be interested in this project!

3.1.1 Project details

Code: https://github.com/mozilla/scrumbugz

Issues: https://github.com/mozilla/scrumbugz/issues

Documentation: http://scrumbugz.readthedocs.org/

IRC: #scrum on irc.mozilla.org

3.1.2 Want to help?

Here are things we need help with:

- **fixing bugs listed in the issue tracker**
- **writing tests**
- **writing documentation**: We could use help writing deployment documentation for deploying Scrumbugz in other environments.
- **spreading the word**: Do you know other people who would like this software? If so, tell them about Scrumbugz!
- **project infrastructure**: Is there infrastructure that's missing in this project that would make it easier for you to collaborate? If so, what?

Are you thinking, "That's a daunting list!" That's ok! Hop on IRC, say hi and we can go from there!

3.2 Hacking HOWTO

3.2.1 Summary

This covers setting up a development environment for Scrumbugz. If you're interested in running Scrumbugz, then you should checkout *Installing Scrumbugz*.

3.2.2 Setup for development

Requirements

- Bugzilla 4+
- The Bugzilla XMLRPC API.

Currently, scrumbugz uses Bugzilla searches for Product(s)/Component(s) and bug IDs via the XMLRPC api (/xml-rpc.cgi).

Thus, in order to use Scrumbugz, you need a Bugzilla instance that's running a recent version of Bugzilla and the Bugzilla API. We think the minimum version is Bugzilla 4, but haven't verified this.

Note: You don't need to install Bugzilla on your machine. As long as you have access to a Bugzilla server, you're fine.

Create virtual environment

Create and activate the virtual environment:

```
virtualenv venv
source venv/bin/activate
```

Note: You don't have to put your virtual environment in `./venv/`. Feel free to put it anywhere.

Get dependencies

Run:

```
pip install -r requirements-dev.txt
```

That sets up all the dependencies required.

Configure

Then you should create a local file. First, copy the template over:

```
cp settings/local.py-dist settings/local.py
```

and edit it.

Set up the db

Run:

```
./manage.py syncdb
./manage.py migrate
```

This also creates a superuser which you can use to log into the Django admin page at http://localhost:8000/admin.

Set up Cache

By default the *settings/local.py* file is set up for a local memory cache. This should be fine for local testing and you shouldn't need to do anything else. If you'd like to more closely mimic production, you can install *memcached* or *Redis* and configure the *CACHES* setting in *settings/local.py* accordingly.

Run it

```
./manage.py runserver
```

Static media will be handled automatically by Django's built-in handler.

3.2.3 Setting up a project

1. Pull up the Django admin page at http://localhost:8000/admin.
2. Login with the admin account you setup during *syncdb*,
3. then go back to the home page at http://localhost:8000/.

Once you're logged in, you'll see buttons for creating and editing projects and sprints. If your superuser account's email address is registered with Mozilla Persona, you can also login using the *Sign In* link on the right of the nav bar.

The Bugzilla url for a sprint should be the url for a query defining the sprint. For example, SUMO uses the target to define sprints, so the query url for our 2012.6 sprint is:

```
https://bugzilla.mozilla.org/buglist.cgi?quicksearch=ALL%20product%3Asupport%20milestone%3A2012.6
```

3.3 Conventions

We follow the code conventions listed in the coding conventions page of the webdev bootcamp guide. This covers HTML, CSS, JS and Python.

We use git and follow the conventions listed in the git and github conventions page of the webdev bootcamp guide.

3.4 Documentation

3.4.1 Conventions

See the docmentation page in the webdev bootcamp guide for documentation conventions.

The documentation is available in HTML and PDF forms at http://scrumbugz.readthedocs.org/. This tracks documentation in the master branch of the git repository. Because of this, it is always up to date.

3.4.2 Building the docs

The documentation in *docs/* is built with Sphinx. To build HTML version of the documentation, do:

```
cd docs/
make html
```

3.5 Tests

3.5.1 Running the tests

To run the tests, do:

```
./manage.py test
```

3.5.2 Writing tests

Tests are located in *scrumb/tests.py*.

They follow Django test conventions.

CHAPTER 4

Indices and tables

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