Mozillians Documentation

Release 0.1

Matthew Riley MacPherson, Giorgos Logiotatidis

Sep 27, 2017

Contents

1 Installation 3
1.1 VirtualEnv Installation . . . 3
1.2 Vagrant Installation . . . 5

2 Testing 7
2.1 Vagrant Box Shortcuts . . . 7
2.2 Test Coverage . . . 7

3 Invitation System 9
3.1 Inviting en-masse . . . 9

4 Indices and tables 11

Mozillians.org is the community phonebook for Mozilla.

This application is built on top of Playdoh.

Contents:

CHAPTER 1

Installation

VirtualEnv Installation

Note: Installing Mozillians might be daunting. Ask for help in #commtools on irc.mozilla.org. Ping *giorgos*, he will be happy to help.

You ‘ll need python, virtualenv and pip.

1. Get a copy of mozillians:

   ```
   $ git clone --recursive git://github.com/mozilla/mozillians.git mozillians
   $ cd mozillians
   ```

2. Create your python virtual environment:

   ```
   $ virtualenv --no-site-packages venv
   ```

3. Activate your python virtual environment:

   ```
   $ source venv/bin/activate
   ```

4. Install development and compiled requirements:

   ```
   (venv)$ pip install -r requirements/compiled.txt -r requirements/dev.txt
   ```

   Note: When you activate your python virtual environment ‘venv’ (virtual environment’s root directory name) will be prepended to your PS1.

   Note: Since you are using a virtual environment all the python packages you will install while the environment is active, will be available only within this environment. Your system's python libraries will remain intact.

5. Configure your local mozillians installation:

   ```
   (venv)$ cp settings/local.py-devdist settings/local.py
   ```

   Note: The provided configuration uses a sqlite database with the filename *mozillians.db* and assumes that server listens to *127.0.0.1:8000*. You can alter the configuration to fit your own needs.

6. Download and run elastic search:

   ```
   (venv)$ wget https://github.com/downloads/elasticsearch/elasticsearch/
   ↪elasticsearch-0.19.4.tar.gz -O /tmp/es.tar.gz
   (venv)$ tar xvf /tmp/es.tar.gz -C venv/
   (venv)$ ./venv/elasticsearch-0.19.4/bin/elasticsearch -p venv/es.pid >/dev/null 2>
   ↪&1
   ```

7. Update product details:

   ```
   (venv)$ ./manage.py update_product_details -f
   ```

8. Sync DB:

   ```
   (venv)$ ./manage.py syncdb --noinput && ./manage.py migrate
   ```

9. Create user:

   (a) Run server:

   ```
   ./manage.py runserver 127.0.0.1:8000
   ```

   (b) Load http://127.0.0.1:8000 and sign in with BrowserID, then create your profile.

   (c) Run:

   ```
   ./scripts/su.sh
   ```

   to vouch your account and convert it to superuser.

10. Develop!

    Now you can start contributing to Mozillians. Check out the Mozillians Git-Fu. When you are done this your coding session, do not forget to kill the *elastic search* process:

    ```
    (venv)$ kill `cat venv/es.pid`
    ```

    and deactivate your virtual python environment by running:

    ```
    (venv)$ deactivate
    ```

    Next time, before starting you will need to start *elasticsearch* server again:

    ```
    $ ./venv/elasticsearch-0.19.4/bin/elasticsearch -p venv/es.pid >/dev/null 2>&1
    ```

    and activate your environment by typing:

```
$ source venv/bin/activate
```

Have fun!

Vagrant Installation

Warning: This documentation needs update. Please use the alternative installation method *VirtualEnv Installation*.

Note: Installing Mozillians might be daunting. Ask for help in #mozillians on irc.mozilla.org. tofumatt or tallOwen will be happy to help.

You'll need ruby, vagrant, Virtualbox and git. The following steps will help you:

1. Install vagrant (requires `ruby`):

   ```
   $ gem install vagrant
   ```

   See also:

   ‘**Vagrant: Getting Started** <http://vagrantup.com/docs/getting-started/index.html>‘_

   Note: you don't need to install the lucid32 box, one will be installed automatically with mozillians)

2. Install virtualbox by Oracle.

   Note: If you run Linux, you'll need to make sure virtualization isn't disabled in your kernel.

3. Get a copy of Mozillians.org:

   ```
   $ git clone --recursive git://github.com/mozilla/mozillians.git mozillians
   $ cd mozillians
   ```

4. Run a virtual dev environment:

   ```
   $ vagrant up
   $ vagrant ssh # you will now enter the virtualized environment
   ```

   Note: Run this in your working copy directory (i.e. `mozillians/`)

   You can edit files under (`mozillians/`) locally and they will automatically show up under /home/vagrant/mozillians in the virtualbox. This means you can edit in your favorite text-editor, yet run Mozillians from our virtualized environment.

5. **Setup the database::** $ ./manage.py update_product_details -f $ mysql -u root > create database mozilians character set utf8; > exit; $ ./manage.py syncdb –noinput

6. Run the development web server (in the virtualized environment):

```
$ ./manage.py runserver 0.0.0.0:8000
```

Note: `rs` is one of the many handy Django aliases included in the Mozillians VM. It's aliases to `./manage.py runserver 0.0.0.0:8000`. You can see all the aliases available by typing `alias` inside your VM shell or by inspecting the contents of `puppet/files/home/vagrant/zshrc` (or `bashrc_vagrant` if you use `bash`).

7. Point your web browser to http://localhost:8000

8. Stay up-to-date:

   On your host machine do:

   ```
   $ git pull -q origin master
   $ git submodule update --recursive
   $ pushd vendor
   $ git pull -q origin master
   $ git submodule update --recursive
   $ popd
   ```

   Then you can run any needed database migrations inside your VM:

   ```
   $ dj syncdb
   $ dj migrate
   ```

   Occassionally there will be a new base VM box. If so, get it with:

   ```
   $ vagrant destroy
   $ vagrant up
   ```

CHAPTER 2

Testing

Vagrant Box Shortcuts

If you are using the *Vagrant Installation* there are a couple of shortcuts to make your life easier:

- Alias *t*:

  ```
  ~4 dj test -x --logging-clear-handlers --with-nicedots'
  ```

- Alias *td*

  ```
  ~$ FORCE_DB=True t --noinput
  ```

- Alias *tf*:

  ```
  ~$ dj test --logging-clear-handlers --with-nicedots --failed
  ```

- Alias *tp*:

  ```
  ~$ tp='t --pdb --pdb-failure'
  ```

Test Coverage

You can combine *nose* testing with the *coverage* module to get the code coverage of the tests. To get a coverage report for the 'users' package run:

```
dj test -x --logging-clear-handlers --with-coverage --cover-package=users
```

You can request to cover multiple packages in one run:

```
dj test -x --logging-clear-handlers --with-coverage --cover-package=users,phonebook
```

CHAPTER 3

Invitation System

Mozillians has an inivitation system that let's vouched users invite others to join Mozillians. These users who join are automatically vouched.

Inviting en-masse

Let's say you have a large list of contributors to invite to your phonebook, well we thought of that.

You can format a file (`myfriends.txt`) with one email address per line:

```
bob@thebobcats.com
juno@reactor.org
diane@hunters.org
```

And feed it on the admin node like so:

```
./manage.py cron invite myfriends.txt
```

And voila! Invitations will be mailed to your friends.

This creates one `Invite` and sets the `receiver` to `ZUUL`. This also sends an invitation email to each recipient.

CHAPTER 4

Indices and tables

- genindex
- modindex
- search