
Between the Bars Documentation

Release 2.0

Between the Bars

Sep 27, 2017

Contents

1	Installation	3
1.1	1. Install system components	3
1.2	2. Set up project directory	4
1.3	3. Clone the repository	5
1.4	4. Set up <code>virtualenv</code> and install python library dependencies	5
1.5	5. Configure BtB	5
1.6	6. Set up database	5
1.7	7. Run the dev server!	6
1.8	8. Set the site name in admin	6
2	Customizing	7
2.1	Site name	7
2.2	Return address	7
2.3	Templates and branding	7
3	Developing	9
3.1	Running the development server	9
3.2	Installing test data	9
3.3	Running tests	9

Between the Bars is a blogging platform for people in prison. People in prison in the US don't generally have access to the Internet, so it's necessary to facilitate their blogging using paper mail.

To that end, BtB has sophisticated tools for managing, moderating, and publishing scanned documents, as well as generating printable batched correspondence. These features alone might make this codebase a compelling alternative for anyone that must process large amounts of hand-written mail, with or without publishing.

This documentation describes how to get started with installing and running a development instance of Between the Bars.

Contents:

1. Install system components

BtB should work on unix systems; it has been tested on Linux and OS X.

System requirements:

- Python 2.6 or 2.7 (Python 3 not supported), and development headers
- `poppler-utils`
- `pdftk`
- `imagemagick`
- `wkhtmltopdf` ≥ 0.10
- `rubber`
- `rabbitmq-server`
- `git` (for obtaining the codebase)
- `mercurial` (for installing python dependencies)
- `python-virtualenv`
- `compass` (for stylesheet compilation)
- `coffee script` (for script compilation)

System-specific instructions:

- Ubuntu:

```
sudo apt-get install git mercurial poppler-utils pdftk imagemagick rubber_
↳rabbitmq-server python-dev postgresql-server-dev-all rubygems nodejs npm_
↳texlive-fonts-extra texlive-fonts-recommended texlive-font-utils texlive-
↳generic-recommended texlive-latex-extra texlive-latex-recommended ttf-sil-
↳gentium

cd /tmp
curl -L https://wkhtmltopdf.googlecode.com/files/wkhtmltopdf-0.11.0_rc1-static-
↳i386.tar.bz2 | tar xjv > wkhtmltopdf
sudo mv wkhtmltopdf /usr/local/bin
sudo gem install compass
sudo npm install -g coffee-script
```

- OS X (using [MacPorts](#) – install that first):

```
sudo port install \
  python26 \
  py26-virtualenv \
  git-core \
  mercurial \
  sqlite3 py26-sqlite \
  poppler \
  pdftk \
  rubber \
  imagemagick \
  rabbitmq-server

# create symlinks to be compatible with other systems
sudo ln -s virtualenv-2.6 /opt/local/bin/virtualenv
sudo ln -sf easy_install-2.6 /opt/local/bin/easy_install

cd /tmp
curl -O https://wkhtmltopdf.googlecode.com/files/wkhtmltopdf-OSX-0.10.0_rc2-
↳static.tar.bz2
tar jxvf wkhtmltopdf-OSX-0.10.0_rc2-static.tar.bz2
sudo cp wkhtmltopdf /usr/local/bin

#XXX: need instructions for installing compass (http://compass-style.org)
#XXX: need instructions for installing coffeescript (http://coffeescript.org)
#XXX: need instructions for installing latex
#XXX: need instructions for installing Gentium font (http://scripts.sil.org/cms/
↳scripts/render\_download.php?&format=file&media\_id=Gentium\_102\_W&
↳filename=Gentium\_102\_W.zip)
```

2. Set up project directory

Set up a directory to put project files/etc in:

```
INSTALL_DIR=~/projects/ # or whatever you like...
mkdir -p $INSTALL_DIR
```

The rest of this documentation assumes there is a `INSTALL_DIR` environment variable.

3. Clone the repository

Download the code:

```
cd $INSTALL_DIR
git clone http://github.com/yourcelf/btb.git btb
```

4. Set up virtualenv and install python library dependencies

Set up a virtualenv into which to install python dependencies. Python dependencies are listed in the file `scanblog/requirements.txt`:

```
VENV_DIR=$INSTALL_DIR/btb/venv # or whatever you like
virtualenv --no-site-packages $VENV_DIR
source $VENV_DIR/bin/activate
easy_install pip
pip install -r $INSTALL_DIR/btb/scanblog/requirements.txt
```

5. Configure BtB

Copy `example.settings.py` to `settings.py`, then edit it to reflect your settings:

```
cd $INSTALL_DIR/btb/scanblog
cp example.settings.py settings.py
```

Be sure to change:

- `ADMINS` and `SERVER_EMAIL` to a suitable name/email
- `TEXT_IMAGE_FONT` to the Gentium font path, e.g., `/usr/share/fonts/gentium/GenR102.TTF`
- Set the path to external executables as appropriate: `NICE_CMD`, `PDFTK_CMD`, `WKHTMLTOPDF_CMD`, `RUBBER_PIPE_CMD`, `PDFINFO_CMD`, `PDFTOTEXT_CMD`, `CONVERT_CMD`
- Change `SECRET_KEY` to something long and random (it's used for hashing authentication cookies).
- If it's a production site, you'll want to use a database other than `sqlite`, as it doesn't support concurrent writes. Set this in the `DATABASES` configuration. (`sqlite` works fine for development)

Due to [this bug](#), as a work around to ensure that images and fonts are served properly in development mode, add symlinks to the compiled asset directory:

```
mkdirs -p $INSTALL_DIR/btb/scanblog/site_static/CACHE
cd $INSTALL_DIR/btb/scanblog/site_static/CACHE
ln -s ../../static/img .
ln -s ../../static/fonts .
```

6. Set up database

Load the initial database, and run initial migrations:

```
source $VENV_DIR/bin/activate
cd $INSTALL_DIR/btb/scanblog

python manage.py syncdb --noinput
python manage.py migrate
python manage.py loaddata btb/fixtures/initial_data.json

# Create superuser
python manage.py shell <<-EOF
from sh import *
u = User.objects.create(username='admin', is_superuser=True, is_staff=True)
u.set_password('admin')
u.save()
exit()
EOF
```

After running that script, there will be a single admin user with username “admin” and password “admin”. This can be changed in the Django admin site by navigating to `http://localhost:8000/admin/`.

7. Run the dev server!

Django ships with a built-in devserver. You can run this directly:

```
cd $INSTALL_DIR/btb/scanblog
source $VENV_DIR/bin/activate
python manage.py runserver
```

8. Set the site name in admin

In order to download documents as PDF’s, you’ll need to set the ‘Site’ object so that it isn’t the default (unless `example.com` resolves as you :)).

To do this, navigate to the admin site: `http://localhost:8000/admin/`. Click `Sites`, and change the default site to a URL that will resolve (probably `localhost:8000`).

CHAPTER 2

Customizing

If you are going to run your own instance of Between the Bars, great! But there are several things you should change first.

Site name

Templates all over the site refer to `{{ site.name }}` – on our site, that’s “Between the Bars”. Set your site’s name using the admin interface.

1. Log in as a superuser or staff user
2. Navigate to `/admin/sites/site/`
3. Change the domain and display name to suitable values.

Return address

Change the default organizational contact and mailing address.

1. Log in as a superuser or staff user
2. Navigate to `/admin/profiles/organization/`
3. Change the name, mailing address, and personal contact of the organization.

Templates and branding

The following templates contain Between the Bars branding and should be modified to suit your organization. Please fix anything

- `scanblog/templates/site_base.html`: change the contact address.

- Everything in `scanblog/templates/about/`, particularly:
 - `index.html`: replace with your own “about” info.
 - `dmca.html`: put your own DMCA contact info in
 - `terms.html`: put terms for your organization, referencing your name
 - `privacy.html`: put a policy for your organization, referencing your name and URL
 - `faq.html`: change authorship, contact address, and other details
 - `guidelines.html`: change contact address.
- `scanblog/templates/home.html`: Remove/change quotes and names to suit your organization. Make yer home page pretty.
- `scanblog/tempaltes/500.html`: put your own URL in.
- Change logos referenced in:
 - `scanblog/templates/site_base.html`
 - `scanblog/templates/home.html`

Terms to grep for in the templates include:

```
grep -Ri between scanblog/templates/  
grep -Ri detar scanblog/templates/  
grep -Ri civic scanblog/templates/  
grep -Ri '\bmit\b' scanblog/templates/
```

Running the development server

Run the development server in the standard way:

```
python manage.py runserver
```

To process scans, you'll need to also run the celery daemon:

```
python manage.py celeryd
```

Note: Previous versions of Between the Bars used a custom wrapper for *runserver* which performed compilation of assets. This is now handled automatically by *django_compressor*.

Installing test data

BtB ships with a small bit of test data to make sure things are working properly. Generate and load this data with the following management command:

```
python manage.py import_test_data
```

The data is described in a YAML file located in `scanblog/media/test/test_data.yaml`.

Running tests

Between the Bars includes unit tests and integration tests. Run unit tests with:

```
python manage.py test
```

Integration tests run with `selenium webdriver`, which runs a real, scripted webbrowser to test the full stack of frontend and backend functionality. Due to the heavy weight of launching and testing with a full browser, these tests are excluded from the default testrunner, and must be run explicitly:

```
python manage.py test btb
```

Note: Some combinations of different selenium webdriver and firefox versions have difficulty with native operations (such as mouse drags) or file uploads. We've found Selenium 2.20 and Firefox 10.0 to be a winning combination. Download an [old firefox binary here](#), and specify the path to this binary in `settings.SELENIUM_FIREFOX_BIN`, e.g.:

```
SELENIUM_FIREFOX_BIN = "/path/to/firefox_10/firefox"
```

Note: Previous versions of Between the Bars used *lettuce* for tests. This has been removed in favor of the standard Django test framework.

Detailed documentation and training materials for how to use Between the Bars in production is forthcoming.