
OpenEats2 Documentation

Release 2.4.0

Quenten Griffith

Jun 27, 2017

Contents

1	Getting Started	3
1.1	Introduction	3
1.2	Installation	3
1.3	Upgrade	7
1.4	Configuration	13
1.5	Comments	14
1.6	Admin Page	15
1.7	Testing	15
1.8	Exporting Recipes	16
1.9	Search Index	16
1.10	Changelog	17
2	Indices and tables	21

This document refers to version 2.4.0

Introduction

OpenEats is a recipe management site that allows users to create, share, and store recipes. OpenEats was created using django, a python web framework and several django plugins. Some of the features of OpenEats are;

- Create, share and edit your own recipes
- Mark your own recipes as private so only you can see them
- Follow other chefs on OpenEats and see what they are rating, and adding
- Create custom grocery list
- Add recipes to your grocery lists
- Share grocery list with other people. This is handy for a family to have a list everyone can update
- Rate recipes
- Comment on recipes
- 100% OpenSource and free
- A complete administration backend to maintain your site
- Export recipes to PDF or MealMaster format
- Internationalization Support

Installation

This document assumes you are familiar with Python and Django.

Requirements

- Python 2.6 or a higher release of 2.x
- Django 1.3.1
- South 0.7.3
- django-disqus 0.4.1
- django-imagekit 0.4.0
- django-taggit 0.9.3
- akismet 0.2.0
- PIL 1.1.7
- django_navbar 0.3.0
- django-profiles 0.2
- feedparser 4.1
- html5lib 0.90
- whoosh 1.8.4
- django-haystack 1.2.4
- django-taggit-templatetags 0.4.6dev
- reportlab 2.5
- django-registration 0.8-alpha-1
- django-relationships 0.3.0
- django-ratings 0.3.6
- django-grappelli 2.3.7

Optional

- webtest 1.2.3 used for testing
- django-webtest 1.4.1 used for testing
- django-sentry 1.10.1 used for debugging and troubleshooting errors
- django-indxer 0.3.0 used by django-sentry
- django-paging 0.2.4 used by django-sentry
- django-extensions 0.6 extra management commands
- django-rosetta 0.6.2 used to ease translations
- django-debug-toolbar 0.8.5 used to help troubleshooting
- django-tastypie 0.9.11
- lxml 2.3

Python Virtual Environment

The easiest way to install OpenEats2, is to create a python virtual environment. This allows for keeping all the packages for OpenEats2 in a separate place. Saving the hassle of dealing with application dependencies.

Installing virtualenv

To install `virtualenv` from the command line type:

```
pip install virtualenv
```

You will also want to install the `virtualenvwrapper` package to make management of the virtual environment simpler

To install `virtualenvwrapper` from the command line type:

```
pip install virtualenvwrapper
```

Creating the virtualenv

To create the skeleton virtualenv run the following commands:

```
export WORKON_HOME=~/Envs
mkdir -p $WORKON_HOME
source /usr/local/bin/virtualenvwrapper.sh
mkvirtualenv openeats --no-site-packages
workon openeats
```

Note: You can set your workon home directory anywhere you want it doesn't have to be in the Envs directory The `virtualenvwrapper.sh` may not be located in `/usr/local/bin` it varies by operating system. For help with `virtualenvwrapper` visit their site.

Installing

- Download the latest package from [github](#)
- Unzip the files into a directory that your web server can access
- Make sure the directory is called `openeats` or python won't be able to find it
- Install the required packages

Installing Requirements packages

To install all the packages that OpenEats2 requires perform the following steps.

- Activate your virtualenv
- Change to the directory that you unzipped the OpenEats2 files into
- Run the following command:

```
pip install -r OE2_Requirements.txt
```

Database

OpenEats2 has been tested with [MySQL](#) and [SQLite](#) and minimal testing has been done with [PostgreSQL](#). Technically it should be able to work under any [django supported](#) database. SQLite is built into python and does not require any additional software.

MySQL

To install the [MySQL-Python](#) module perform the following steps

- Activate your OpenEats2 virtualenv
- Run the following command:

```
pip install mysql-python
```

PostgreSQL

To install the [Postgres](#) module perform the following steps

- Activate your OpenEats2 virtualenv
- Run the following command:

```
pip install psycopg2
```

There is a small issue with PostgreSQL that will cause you an error when loading the data. To get around this issue perform the following steps;

- Copy the `postgres_settings.py` to `settings.py`.
- Then skip running the migrate command from the *Required Data* section below.
- After running the `./manage.py syncdb` command from the *Required Data* section, edit the `settings.py` file and remove the `#` from in front of the word `south` in the file and save it.
- Then run the command `./manage.py migrate --fake`.
- Continue with the rest of the instructions as normal.

Load Initial Data

OpenEats2 comes with default data that needs to be loaded into the database.

Required Data

Running the following command from the OpenEats2 directory, should load the required data:

```
./manage.py syncdb --all
./manage.py migrate --fake
./manage.py loaddata fixtures/navbar_about_data.json
```

Note: Before you run this make sure you have setup your database in the `settings.py` file. For more information on this see [Database](#)

Optional Data

You can pre-load courses and cuisines by running the following commands from the OpenEats2 directory:

```
./manage.py loaddata recipe_groups/fixtures/course_data.json
./manage.py loaddata recipe_groups/fixtures/cuisine_data.json
```

Collecting Static Files

To collect the static files from the third party applications run the following command:

```
./manage.py collectstatic
```

Running

After the install you can run the following command to start the internal Django webserver. This will allow you to test your site prior to setting up a “real” webserver such as Apache:

```
./manage.py runserver 8000
```

This will bind the webserver to port 8000 on 127.0.0.1 otherwise known as localhost. If you are deploying OpenEats2 to a remote server and not your local computer run the following command instead:

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

You should then be able to access your new OpenEats2 site by pointing your browser to your URL with port 8000:

```
http://yoursite:8000
```

Note: You should not run OpenEats2 in production with the built in webserver. You will want to setup [Apache](#) or [Nginx](#) Check out the [Django Apache WSGI](#) document for more info.

Site Name

You will need to set up your site name before you can use certain features. See [Change Site Name](#)

Upgrade

Upgrade 2.0 to 2.1

This procedure covers how to upgrade your existing 2.0 OpenEats2 site to version 2.1

Install

- Download the latest package from [github](#)
- Unzip the file

- Backup your database
- Backup your current site files
- From the unzip file, copy everything other than the settings.py, site-media/upload and site-media/uploads file into your current openeats directory, over writing what is already there

Config File Changes

Some minor changes need to be made to the settings.py file of your current site.

Remove the following from the installed apps section of the settings.py file

- admin_tools.theming
- admin_tools.menu
- admin_tools.menu

Add the following to the installed apps section of the settings.py file on the *first* line:

```
'grappelli.dashboard',
```

Add the following to the installed apps section of the settings.py file:

```
'django.contrib.staticfiles',
```

Remove the following from the *context processors* section of the settings.py file:

```
"grappelli.context_processors.admin_template_path",
```

Add the following to the *context processors* section of the settings.py file:

```
"openeats.context_processors.oetitle",
```

Add the following line to the settings.py file:

```
LOCALE_PATHS = (  
    os.path.join(BASE_PATH, 'locale',)  
)
```

Remove the following lines in your settings.py file:

```
ADMIN_TOOLS_INDEX_DASHBOARD = 'openeats.dashboard.CustomIndexDashboard'  
ADMIN_TOOLS_APP_INDEX_DASHBOARD = 'openeats.dashboard.CustomAppIndexDashboard'
```

Add the following line to your settings.py file:

```
GRAPPELLI_INDEX_DASHBOARD = 'openeats.dashboard.CustomIndexDashboard'
```

Add the following lines to your settings.py file:

```
STATIC_ROOT = os.path.join(BASE_PATH, 'static-files')  
STATIC_URL = '/static-files/'
```

Change the *ADMIN_MEDIA_PREFIX* setting in the settings.py file to this:

```
ADMIN_MEDIA_PREFIX = STATIC_URL + "grappelli/"
```

Add the following line under the *OpenEats Settings* section of the settings.py file:

```
OETITLE="OpenEats2"
```

You can set this to anything you want your site title to be, it doesn't have to be set to *OpenEats2*

Remove the following line from urls.py:

```
url(r'^admin_tools/', include('admin_tools.urls')),
```

Removing directories

The following directories are no longer needed for version 2.1 and can be removed

- grappelli
- site-media/admin
- site-media/admin_tools

Upgrade third party apps

To upgrade all the packages that OpenEats2 requires perform the following steps.

- Activate your virtualenv
- Change to the directory that you unzipped the OpenEats2 files into
- Run the following command:

```
pip install -r OE2_Requirements.txt --upgrade
```

Database changes

Run the following command from the OpenEats2 directory:

```
./manage.py syncdb
./manage.py migrate djangoratings 0001 --fake
./manage.py migrate djangoratings 0002 --fake
./manage.py migrate djangoratings 0003
./manage.py migrate djangoratings 0004 --fake
./manage.py migrate djangoratings 0005 --fake
./manage.py migrate djangoratings
./manage.py migrate sentry 0001 --fake
./manage.py migrate sentry 0002 --fake
./manage.py migrate sentry 0003 --fake
./manage.py migrate sentry 0004 --fake
./manage.py migrate sentry 0005 --fake
./manage.py migrate sentry 0006 --fake
./manage.py migrate sentry 0007 --fake
./manage.py migrate sentry 0008 --fake
./manage.py migrate sentry 0009 --fake
./manage.py migrate sentry
./manage.py migrate
```

Third Party static files

Django 1.3 offers a new feature that pulls in all the css and image files from third party apps into one folder. This does away with having to setup symlinks to them yourself. The files are stored in the static-files directory. To get the initial files set run the following command:

```
./manage.py collectstatic
```

Rebuild Search Index

The search engine was updated as part of this release so it is a good idea to run the following command to rebuild it:

```
./manage.py rebuild_index
```

Running

After the upgrade you can run the following command to start the internal Django webserver. This will allow you to test your site:

```
./manage.py runserver 8000
```

Upgrade 2.1 to 2.2

This procedure covers how to upgrade your existing 2.1 OpenEats2 site to version 2.2. If you are upgrading from 2.0 to 2.2 you will need to follow the steps from Config File Changes down in the *Upgrading from 2.0 to 2.1* section.

Install

- Download the latest package from [github](#)
- Unzip the file
- Backup your database
- Backup your current site files
- From the unzip file, copy everything other then the settings.py, site-media/upload and site-media/uploads file into your current openeats directory, over writing what is already there

Database changes

Run the following command from the OpenEats2 directory:

```
./manage.py migrate list
```

Running

After the upgrade you can run the following command to start the internal Django webserver. This will allow you to test your site:

```
./manage.py runserver 8000
```

Upgrade 2.2 to 2.3

This procedure covers how to upgrade your existing 2.2 OpenEats2 site to version 2.3. If you are upgrading from 2.0 to 2.3 you will need to follow the steps from Config File Changes down in the *Upgrading from 2.0 to 2.1* section.

Install

- Download the latest package from [github](#)
- Unzip the file
- Backup your database
- Backup your current site files
- From the unzip file, copy everything other than the settings.py, site-media/upload and site-media/uploads file into your current openeats directory, over writing what is already there

Settings Changes

Add the following to your settings.py file under the TEMPLATE_CONTEXT_PROCESSORS area:

```
'django.core.context_processors.static',
```

Remove the following from the installed_apps section of the settings.py file:

```
'reversion',
```

Upgrade third party apps

To upgrade all the packages that OpenEats2 requires perform the following steps.

- Activate your virtualenv
- Change to the directory that you unzipped the OpenEats2 files into
- Run the following command:

```
pip install -r OE2_Requirements.txt --upgrade
```

Database changes

Run the following command from the OpenEats2 directory:

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

Update Static Files

To update the static files from the third party applications run the following command:

```
./manage.py collectstatic
```

Running

After the upgrade you can run the following command to start the internal Django webserver. This will allow you to test your site:

```
./manage.py runserver 8000
```

Upgrade 2.3 to 2.4

This procedure covers how to upgrade your existing 2.3 OpenEats2 site to version 2.4. If you are upgrading from 2.0 to 2.4 you will need to follow the steps from Config File Changes down in the *Upgrading from 2.0 to 2.1* section.

Install

- Download the latest package from [github](#)
- Unzip the file
- Backup your database
- Backup your current site files
- From the unzip file, copy everything other than the settings.py, site-media/upload and site-media/uploads file into your current openeats directory, over writing what is already there

Settings Changes

Remove the following from the installed_apps section of the settings.py file:

```
'django_generic_flatblocks',
```

Upgrade third party apps

To upgrade all the packages that OpenEats2 requires perform the following steps.

- Activate your virtualenv
- Change to the directory that you unzipped the OpenEats2 files into
- Run the following command:

```
pip install -r OE2_Requirements.txt --upgrade
```

Database changes

Run the following command from the OpenEats2 directory:

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

Update Static Files

To update the static files from the third party applications run the following command:

```
./manage.py collectstatic
```

Update Recipe Picture cache

The size of the recipe pictures has changed to re-create the current pictures to the new size run:

```
./manage.py ikflush recipe
```

Running

After the upgrade you can run the following command to start the internal Django webserver. This will allow you to test your site:

```
./manage.py runserver 8000
```

Configuration

This document covers the configuration settings for OpenEats2

Database

To setup Django for your database refer to the [Django Database](#) document

Email

To setup your email server open up the settings.py file located in the OpenEats2 directory and fill in the following settings with your own email server settings:

```
#Email Server Settings
DEFAULT_FROM_EMAIL = ''
EMAIL_HOST = ''
EMAIL_HOST_USER = ''
EMAIL_HOST_PASSWORD = ''
EMAIL_PORT =
#EMAIL_USE_TLS = True
```

Note: If you do not have your own email server you can use Gmail if you have a gmail account. Follow this [gmail guide](#) to learn how.

Logo

If you want to change the logo to your own logo you can modify the OELOGO setting in the setting.py file:

```
#OpenEats2 Settings
OELOGO = 'images/oelogo.png'
```

Change the oelogo.png to what ever your logo image name is and place your new logo in *site-media/images* directory

Site Title

If you want to set your own site title, different from the default OpenEats2. You can modify the OETITLE setting in the setting.py file:

```
OETITLE = 'OpenEats2'
```

Change the word between the quotes to what ever you want your site title to be

Debug

By default OpenEats2 is in debug mode to allow users to find issues with their site during initial setup. Once conferable with your site you should turn debug mode off by changing the following line in your settings.py file and restarting your webserver:

```
DEBUG = True

change to

DEBUG = False
```

Comments

OpenEats2 uses [Disqus](#) for its comment system. In order to use Disqus on your OpenEats2 site you will need to apply for an API key and register your site.

Disqus API Key

To get your own Disqus API key visit the [Disqus register](#) site and sign up. Make sure you enter the fully qualified domain name of where you are hosting OpenEats2.

Disqus Configuration

Once you have register your site with Disqus you will need to add the following to your settings.py file:

```
#Disqus settings
DISQUS_API_KEY = "your API key"
DISQUS_WEBSITE_SHORTNAME = "the shortname you entered when you registered"
```

Admin Page

OpenEats2 has an Admin backend to allow the administrator of the site manager their OpenEats2 site.

Access Admin Site

To access the Admin site point your browser to the following URL:

```
http://yourdomain/admin
```

You will be prompted to login with a userid and password the defaults are:

```
user: admin
password: password
```

You will want to change this as soon as you login by click on the *users* link inside the admin page

Change Site Name

You will want to change the site name on the Admin page to match your domain name. To do so click on the *sites* link on the Admin Page then click on the current entry for site name which should be *example* then change that to match your domain name, with out the *http or www* part

Note: Do not just delete the name that is currently there. Django expects your site name to start with the primary key of 1 in the database. If you delete the site name and add a new one it will have the id of 2. So edit the site name that is already there.

Testing

OpenEats2 comes with several built in test to make sure the code is working as designed. To run the test make sure you have installed the following packages in your virtuenv

- `webtest` 1.2.3 used for testing
- `django-webtest` 1.2.1 used for testing

Running Test

To run the test run the following command once you are in your virtuenv:

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

recipe can be replaced with which ever app you want to test. Currently the following apps have tests

- recipe
- accounts
- list

Exporting Recipes

You can export individual recipes or every recipe you have on your site into the common MealMaster format.

Exporting Individual Recipes

To export individual recipes login to the Admin part of the site [Admin Page](#). Click on the *Recipes* link. Then place a check mark in the box next to each recipe you want to export. At the bottom of the page there is a dropdown from the dropdown select *Export MealMaster* A file will be downloaded to your computer with the exported recipes.

Export all Recipes

To export all the recipes on your site to the MealMaster format, you will need to run a command from the directory you installed OpenEats2. Activate your virtualenv, see [Creating the virtualenv](#) then run the following command from the OpenEats2 directory:

```
./manage.py export_recipes > recipe.txt
```

This will create a file called `recipe.txt` which will contain all the recipes on your site in the MealMaster format

Search Index

The search feature on OpenEats2 is handled by two different applications; [django-haystack](#) and [whoosh](#)

Updating the Index

You have two options when updating the search index;

- Real time, as a recipe is added or deleted the index is updated, this is the default method.
- Scheduled, you can run a script every x minutes to update the index

Real time Updates

By default OpenEats2 ships with real time update dates out of the box; meaning if a recipe is added the search index is updated. There are some downsides to this. If you have a real active site, updating the index in real time could cause a huge slow down. The handoff between the database save and the search index update could cause some weird issues in your database such as items only being half saved, this would happen during a heavy load on the system.

Note: If you are going to run OpenEats2 on Apache with `mod_wsgi`, you will need to enable `mod_wsgi` to run in daemon mode for real time updates. Otherwise everyone once in a while you will get an error that the index can't be updated because of garbage collection on Apache.

Schedule Updates

If you do not wish to run real time updates you can set the index to be update via a script that you run on a schedule. To do this you will want to edit the `recipe/search_indexes.py` file:

```
Change line

class RecipeIndex(RealTimeSearchIndex):

to

class RecipeIndex(SearchIndex):
```

After the change you will need to restart your web server. An example of a script that will update the index can be found in the `contrib` folder of your OpenEats2 project. With out the script you can run the following commands from your `virtenv`:

```
./manage.py rebuild_index *this command creates the index
./manage.py update_index *this updates an existing index
```

Changelog

Version 2.1

- Upgraded Django to 1.3
- Added CSRF protection for AJAX calls
- Upgraded to django-grappelli 2.3.3
- Removed grappelli from the root dir of OpenEats, it now installs like all the ohter thrid party apps
- Removed `admin_tools` third party app
- Setup grappelli dashboard to replace the `admin_tools` dashboard
- Setup static files a new feature in Django 1.3. This pulls in all the third party apps images, css files. replaces the need to do symnlinks to them
- Upgraded django-imagekit to 0.3.6
- Upgraded django-debug-toolbar to 0.8.5
- Upgraded south to 0.7.3
- Upgraded django-haystack to 1.2.4
- Upgraded Whoosh to 1.8.4
- Upgraded django-rosetta to 0.6.2
- Upgraded django-sentry to 1.8.6.2

- Upgraded django-webtest to 1.4.1
- Upgraded django-relationships to 0.3.0
- Upgraded django-ratings to 0.3.6
- Added the ability for users to change the tile of the site via settings.py
- Added the ability to change and clear a picture on a recipe when edited

Version 2.2

- Added meta keywords and description to the main page
- Added meta keywords using recipe titles and tags to the recipe detail page
- Fixed bug of ratings not showing up on a recipe till a user voted on them even if another user had rated the recipe
- Added a top recipe view that shows the top 20 recipes
- Added a most recent recipes view that shows the 20 most recent recipes
- Added recent recipe rss feed
- Added user created aisles for grocery lists
- Added recipes displayed on grocery list when they are added to a list
- Added django-tastypie package for API
- Fixed urls.py by removing a url to a package that is not not used anymore

Version 2.3

- Added instructions and a special settings file to support PostgreSQL
- Moved the api module to the contrib folder
- Sorted grocery list and shared grocery list by created date
- Changed the submit button text on all forms to make more sense
- Changed the design of the title bar that is found above all the category list of recipes
- Removed the “follow me” link from a users profile if they are viewing their own profile.
- Branched the django ratings application and updated it to work with django 1.3
- Lots of CSS tweaks
- Upgraded to Django 1.3.1
- Moved all the CSS, images, and JS to a new folder called static to be more in-line with the Django 1.3 way of doing things
- Upgraded Grapelli to version 2.3.4
- Upgraded django-generic-flatblocks to 0.9.1
- Removed the plugin django-reversion
- Upgraded django-imagekit to 0.4.0
- Changed the site Logo

Version 2.4

- Added a new news section now you can have more than one news story and decide which one appears on the front page
- Added the ability to email recipes to friends
- Added grocery list API
- Upgraded django-tastypie
- Complete UI overhaul using Twitter Bootstrap
- Changed image size for recipe pictures
- Upgraded Grapelli to version 2.3.7
- Removed un-needed java script libraries
- Code clean up
- several small bug fixes

CHAPTER 2

Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)