

---

# **django-planet Documentation**

*Release 0.5.1*

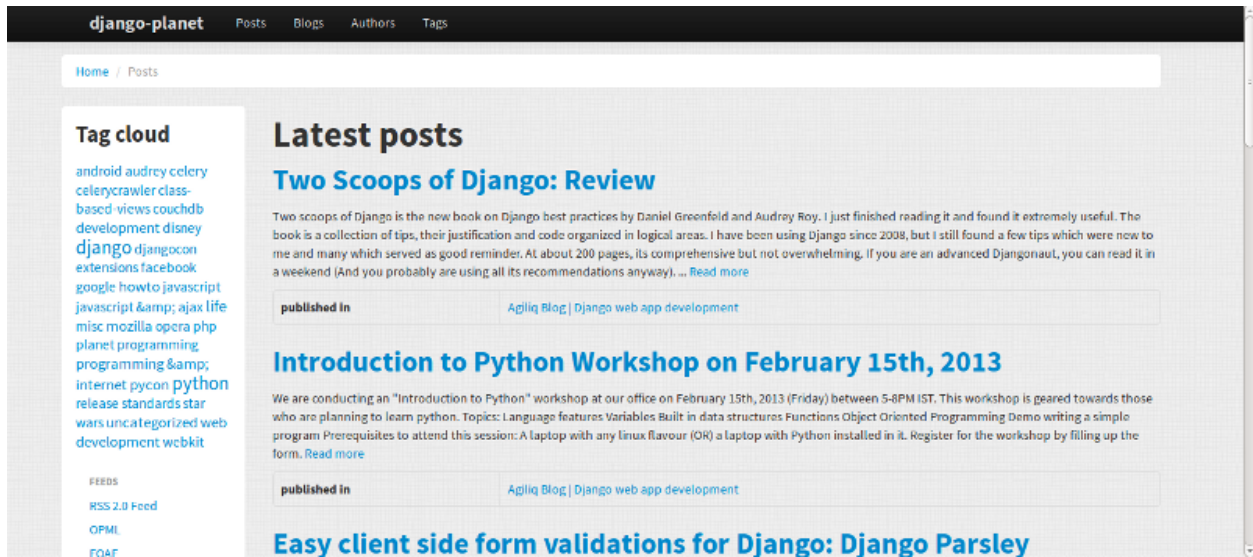
**Matías Agustín Méndez**

September 21, 2016



<b>1</b>	<b>Content:</b>	<b>3</b>
<b>2</b>	<b>Getting Help</b>	<b>13</b>
<b>3</b>	<b>Requirements</b>	<b>15</b>
<b>4</b>	<b>Why django-planet?</b>	<b>17</b>
<b>5</b>	<b>Running The Tests</b>	<b>19</b>
<b>6</b>	<b>Indices and tables</b>	<b>21</b>





This is a generic application for Django that allows you to quickly build a planet aggregating RSS and ATOM feeds of your favorite blogs.

Some parts of this help docs has been copied from [django-tastypie](#) and then readapted to django-planet. Kudos to [django-tastypie](#) for its docs!



## 1.1 Usage

### 1.1.1 Changing your settings.py

Modify your projects `settings.py` file following the next steps:

1. Check your `INSTALLED_APPS`:

```
INSTALLED_APPS = (  
    # django required contrib apps  
    'django.contrib.sites',  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'django.contrib.sitemaps',  
    # 3rd-party required apps:  
    'pagination',  
    'tagging',  
    'pinax_theme_bootstrap',  
    # and finally:  
    'planet',  
)
```

2. Configure your database. Here is an example using `mysql`:

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql', # Add 'postgresql_psycopg2', 'postgresql', 'mysql', 's  
        'NAME': 'planet', # Or path to database file if using sqlite3.  
        'USER': '<myuser>', # Not used with sqlite3.  
        'PASSWORD': '<mypass>', # Not used with sqlite3.  
        'HOST': '', # Set to empty string for localhost. Not used with sqlite3.  
        'PORT': '', # Set to empty string for default. Not used with sqlite3.  
    }  
}
```

3. Choose a site id:

```
SITE_ID = 1
```

#### 4. For Django 1.8 include the following context processors:

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [
            '/path/to/project/templates',
        ],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
                'django.template.context_processors.i18n',
                'django.template.context_processors.media',
                'django.template.context_processors.static',
                'django.template.context_processors.tz',
                'planet.context_processors.context',
            ],
        },
    },
]
```

If you're still using Django 1.6.x or 1.7.x, then set `TEMPLATE_CONTEXT_PROCESSORS` this way:

```
TEMPLATE_CONTEXT_PROCESSORS = (
    'django.contrib.auth.context_processors.auth',
    'django.core.context_processors.debug',
    'django.core.context_processors.i18n',
    'django.core.context_processors.media',
    'django.core.context_processors.static',
    'django.core.context_processors.tz',
    'django.core.context_processors.request',
    'django.contrib.messages.context_processors.messages',
    'planet.context_processors.context',
)
```

#### 5. Check your middlewares to include:

```
MIDDLEWARE_CLASSES = (
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
    'pagination.middleware.PaginationMiddleware',
)
```

Please do not forget `pagination.middleware.PaginationMiddleware` middleware!

#### 5. Add planet configuration variables:

```
PLANET = {
    "USER_AGENT": "My Planet/1.0",
}
```



- Properly configure your static files root directory:

```
STATIC_URL = '/static/'
```

- Only for Django 1.6.x or 1.7.x set your projects templates root directory:

```
TEMPLATE_DIRS = (
    '/path/to/project/templates',
    # other paths...
)
```

and your template loaders:

```
TEMPLATE_LOADERS = (
    'django.template.loaders.filesystem.Loader',
    'django.template.loaders.app_directories.Loader',
    # some other template loaders here...
)
```

- Finally in your project's templates directory create a `site_base.html` template if you don't already have one:

```
{% extends "base.html" %}
```

- Optionally, modify cookie names so you don't have login conflicts with other projects:

```
LANGUAGE_COOKIE_NAME = "myplanetlng"
SESSION_COOKIE_NAME = "myplanetid"
```

Congratulations! Your settings are complete. Now you'll need to change other files in order to get a running project.

### 1.1.2 Enable planet urls

- Add the planet urls include to your project's `urls.py` (remember to also include admin urls so you can use the admin to manage your planet!):

```
from django.conf.urls import patterns, include, url

from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
    url(r'^$', include('planet.urls')),
    url(r'^admin/', include(admin.site.urls)),
    # ... other url bits...
)
```

### 1.1.3 Syncdb and add some feeds!

- Then create the database structure:

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

- Add some feeds:

```
python manage.py planet_add_feed http://www.economonitor.com/feed/rss/
python manage.py planet_add_feed http://www.ft.com/rss/home/us
```

3. And surely you'll want to add a cron entry to periodically update them all:

```
30 * * * * python manage.py planet_update_all_feeds
```

This attempts to pull in new posts every 30 minutes.

4. Now you're done. Just run:

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

and browse your planet at <http://localhost:8000/> in your favorite browser!

## 1.2 How does it look like?

The following screenshots are just for demonstration purposes only:



## 1.2.1 Index page

The screenshot shows the homepage of django-planet. At the top, there is a navigation bar with 'django-planet' and links for 'Posts', 'Blogs', 'Authors', and 'Tags'. Below the navigation bar, there is a breadcrumb trail: 'Home / Posts'. On the left side, there is a 'Tag cloud' section with a list of tags including 'android', 'audrey', 'celery', 'celerycrawler', 'class-based-views', 'couchdb', 'development', 'disney', 'django', 'djangocon', 'extensions', 'facebook', 'google', 'howto', 'javascript', 'javascript & ajax', 'life', 'misc', 'mozilla', 'opera', 'php', 'planet', 'programming', 'programming & internet', 'pycon', 'python', 'release', 'standards', 'star wars', 'uncategorized', 'web', 'development', and 'webkit'. Below the tag cloud are links for 'FEEDS', 'RSS 2.0 Feed', 'OPML', and 'FOAF'. The main content area is titled 'Latest posts' and features two article previews. The first preview is for 'Two Scoops of Django: Review' by Daniel Greenfeld and Audrey Roy, published in 'Agliq Blog | Django web app development'. The second preview is for 'Introduction to Python Workshop on February 15th, 2013', published in 'Agliq Blog | Django web app development'. Below these previews is a third preview for 'Easy client side form validations for Django: Django Parsley'.

## 1.2.2 Post view

The screenshot shows a detailed view of a post on django-planet. The navigation bar at the top is the same as in the index page. The breadcrumb trail is 'Home / Blogs / Between engineering and real life / Posts / Introduction to Django - presentation'. On the left side, there is a 'Tags cloud for feed Between engineering and real life' section with tags for 'development', 'django', 'education', 'python', 'web', and 'web development'. Below the tag cloud are links for 'FEEDS', 'RSS 2.0 Feed', 'OPML', and 'FOAF'. The main content area is titled 'Introduction to Django - presentation' and is dated 'Mar 16 2013'. It includes a 'published by' field with the name 'Ilian Iliev', an 'in blog' field with 'Between engineering and real life', and an 'original entry' field with 'Introduction to Django - presentation'. There are also tags for 'django' and 'education'. The post content begins with 'This presentation shows the basics of Django - what is inside the framework and explains the Model-View-Template system. One of the most important parts is a diagram how the request is processed and the response is generated. Shows the project and the application structure and the basic elements - [...]'. At the bottom right of the page, there is a footer that says 'A django planet powered site'.

## 1.2.3 Tag view

The screenshot shows a tag view on django-planet. The navigation bar at the top is the same as in the index page. The breadcrumb trail is 'Home / Tags / django'. On the left side, there is a 'RELATED TAGS' section with a list of tags including '1.0', '1.0.1', '1.0.2', '1.0.3', '1.1.0', '1.1.1', 'admin', 'adsense', 'ajax', 'api', 'app', 'apple', and 'audrey'. The 'admin' tag is highlighted. The main content area is titled 'Tag: django' and is dated 'Mar 18 2013'. It features a preview of the 'Introduction to Django - presentation' post, including its title, tags ('django', 'education'), and a 'published in' field with 'Between engineering and real life'. Below this preview is another preview for 'Setting up a Django Server with PostgreSQL, Nginx, Celery & RabbitMQ', dated 'Mar 13 2013'. At the bottom of the page, there is a footer with a disclaimer: 'Disclaimer: I am not a sysadmin. I'm just a developer. I welcome and encourage comments and questions. I have set up a couple of Django servers lately and taken copious notes that I have extracted from various sources. Below are the commands I use to a fresh Ubuntu server install to get Django [...] Related posts: Setting Up Your Own Sentry Server if you are hosting a Django site, Sentry will make... Django Custom Upload'.

## 1.3 Demo Project

## 1.4 Sites Using django-planet

The following sites are a partial list of people using django-planet. I'm always interested in adding more sites, so please find me ([matagus](#)) via twitter or start a mailing list thread.

### 1.4.1 django-planet demo website:

Running on [Heroku](#):

- Site: <https://django-planet.com/>

## 1.5 Contributing

django-planet is open-source and, as such, grows (or shrinks) & improves in part due to the community. Below are some guidelines on how to help with the project.

### 1.5.1 Philosophy

- django-planet is BSD-licensed. All contributed code must be either
  - the original work of the author, contributed under the BSD, or...
  - work taken from another project released under a BSD-compatible license.
- GPL'd (or similar) works are not eligible for inclusion.
- django-planet's git master branch should always be stable, production-ready & passing all tests.

### 1.5.2 Guidelines For Reporting An Issue/Feature

So you've found a bug or have a great idea for a feature. Here's the steps you should take to help get it added/fixd in Tastypie:

- First, check to see if there's an existing issue/pull request for the bug/feature. All issues are at <https://github.com/matagus/django-planet/issues> and pull reqs are at <https://github.com/matagus/django-planet/pulls>.
- If there isn't one there, please file an issue. The ideal report includes:
  - A description of the problem/suggestion.
  - How to recreate the bug.
  - If relevant, including the versions of your:
    - \* Python interpreter
    - \* Django
    - \* django-planet
    - \* Optionally of the other dependencies involved
  - Ideally, creating a pull request with a (failing) test case demonstrating what's wrong. This makes it easy for us to reproduce & fix the problem. Instructions for running the tests are at [Welcome to django-planet!](#)

### 1.5.3 Guidelines For Contributing Code

If you're ready to take the plunge & contribute back some code/docs, the process should look like:

- Fork the project on GitHub into your own account.
- Clone your copy of django-planet.
- Make a new branch in git & commit your changes there.
- Push your new branch up to GitHub.
- Again, ensure there isn't already an issue or pull request out there on it. If there is & you feel you have a better fix, please take note of the issue number & mention it in your pull request.
- Create a new pull request (based on your branch), including what the problem/feature is, versions of your software & referencing any related issues/pull requests.

In order to be merged into django-planet, contributions must have the following:

- A solid patch that:
  - is clear.
  - works across all supported versions of Python/Django.
  - follows the existing style of the code base (mostly PEP-8).
  - comments included as needed.
- A test case that demonstrates the previous flaw that now passes with the included patch.
- If it adds/changes a public API, it must also include documentation for those changes.
- Must be appropriately licensed (see "Philosophy").
- Adds yourself to the AUTHORS file.

If your contribution lacks any of these things, they will have to be added by a core contributor before being merged into django-planet proper, which may take substantial time for the all-volunteer team to get to.

### 1.5.4 Guidelines For Core Contributors

If you've been granted the commit bit, here's how to shepherd the changes in:

- Any time you go to work on djano-planet, please use `git pull --rebase` to fetch the latest changes.
- Any new features/bug fixes must meet the above guidelines for contributing code (solid patch/tests passing/docs included).
- Commits are typically cherry-picked onto a branch off master.
  - This is done so as not to include extraneous commits, as some people submit pull reqs based on their git master that has other things applied to it.
- A set of commits should be squashed down to a single commit.
  - `git merge --squash` is a good tool for performing this, as is `git rebase -i HEAD~N`.
  - This is done to prevent anyone using the git repo from accidentally pulling work-in-progress commits.
- Commit messages should use past tense, describe what changed & thank anyone involved.
  - “““Added template tag to get all posts.
  - Further description appears here if the change warrants an explanation as to why it was done.”““

- For any patches applied from a contributor, please ensure their name appears in the AUTHORS file.
- When closing issues or pull requests, please reference the SHA in the closing message (i.e. Thanks! Fixed in SHA: 6b93f6). GitHub will automatically link to it.





---

### Getting Help

---

There are two primary ways of getting help. We have a [mailing list](https://groups.google.com/forum/#!forum/django-planet) hosted at Google (<https://groups.google.com/forum/#!forum/django-planet>) or you may contact us via email to matagus at gmail dot com. You may also open an issue in our github repository (it requires you to have a github account).



---

## Requirements

---

django-planet requires the following modules but simply installing it using [Pip](#) will also install them. Just type:

```
pip install django-planet
```

### 3.1 Required

- Python 3.4+ or 2.7+
- Django 1.6 or 1.7 or 1.8
- django-tagging 0.3.6
- django-pagination 1.0.0+ for python 2.7 or django-pagination-py3 for python 3.5
- feedparser >= 5.0
- pinax-theme-bootstrap >= 3.0
- BeautifulSoup4 >= 4.0

Optionally, install celery if you want to add and update feeds using async & parallel tasks:

- Celery >= 3.0
- django-celery >= 3.0

### 3.2 Optional

- south (only required if you're using Django 1.6.x)



---

## Why django-planet?

---

There are other feed aggregators out there for Django. You need to assess the options available and decide for yourself. That said, here are some common reasons for django-planet.

- You need to quickly create a blog aggregator website with a nice look & feel.
- You want a full website for browsing blog posts and its authors and tags, feeds and blogs.
- SEO matters to you: django-planet has templates with SEO metatags and it includes sitemaps so you may submit them to your favorite search engines.
- You want searching posts, blogs, tags and authors.
- You need to customize templates and have a rich set of template tags to do it.
- You want complete ATOM & RSS support



---

## Running The Tests

---

The easiest way to get setup to run django-planet's tests looks like:

```
$ git clone https://github.com/matagus/django-planet.git
$ cd django-planet
$ virtualenv env
$ . env/bin/activate
$ ./env/bin/pip install -U -r requirements.txt
$ ./env/bin/pip install -U mock django-discover-runner factory-boy tox
```

Then running the tests is as simple as:

```
# From the same directory as above:
$ tox
```

That will test django-planet using Python 2.7 combined with Django 1.4, Django 1.5 and Django 1.6.





---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`