Django Andablog Documentation

Release 3.0.0

Ivan Ven Osdel

Apr 15, 2020

Contents

A blog app that is only intended to embed within an existing Django site.

- Free software: BSD license (source)

- Compatible with 2 most recent Django versions (see history to confirm if we are keeping up)

- Compatible with Python 3.x and 2.7

Getting Started

1.1 Installation & Usage

The easiest way to install Andablog is with pip; this will give you the latest version available on PyPi:

```
pip install django-andablog
```

If you are adventurous (or we are just slow) you can get the latest code directly from the Github repository:

```
pip install -e git+https://github.com/WimpyAnalytics/django-andablog.git#egg=django-
↪andablog
```

The master branch can generally be considered bug free though newer features may be a little half baked. For more information see the official Python package installation tutorial.

1.1.1 Django Settings

1. Check Django pre-requisites

- Confirm that your site's MEDIA_ROOT and MEDIA_URL settings are correct.

- Django's site framework should be enabled.

- The Django admin should be enabled if you wish to use the pre-canned blog administration tools

2. Add to your INSTALLED_APPS:

```
INSTALLED_APPS = (
    # ... other applications,
    'andablog',
    'taggit',  # For entry tags
    'south',   # Only if your site is on Django 1.6
)
```

3. Run the migrations:

```
$ python manage.py migrate
```

4. (Optional) Configure andablog to use a markup syntax for blog entries.

   For Markdown, install the Markdown pypi package and add the appropriate Markupfield! settings to your settings.py

1.1.2 Integrating Andablog into a Site

The following tasks allow for all possible andablog features. Ignore the items you don't need.

Included Pages

To use the pages provided by andablog add something like this to your site's URL hierarchy:

```
(r'^blog/', include('andablog.urls', namespace='andablog')),
```

Then modify your site's navbar to link to the blog listing. E.g.

   Blog

Finally, override andablog's base template to inherit from your site's base.html.

   andablog/base.html

Note: The andablog templates make no assumptions when it comes to the content of your site's template. All blocks referenced by andablog are prefixed by 'andablog' and you place them how you like.

The demo app has an example of overriding andablog's base.html.

Blog Entry Comments

Andablog can use contrib comments or any other pluggable commenting system (such as your own).

To provide andablog with comments, override the following template snippets:

```
andablog/comments_count_snippet.html
andablog/comments_snippet.html
```

Comments using Disqus

Disqus is a service which provides commenting plug-in as a JavaScript and <iframe> embed for any HTML system. Disqus has free and paid plans.

To use Disqus with Andablog, sign up on Disqus to get your id, add and modify the following ndablog/comments_snippet.html example:

```
<div id="disqus_thread"></div>

<script type="text/javascript">
    /* * * CONFIGURATION VARIABLES * * */
    var disqus_shortname = 'YOURIDGOESHERE';
```

```
    /* * * DON'T EDIT BELOW THIS LINE * * */
    (function() {
        var dsq = document.createElement('script'); dsq.type = 'text/javascript'; dsq.
→async = true;
        dsq.src = '//' + disqus_shortname + '.disqus.com/embed.js';
        (document.getElementsByTagName('head')[0] || document.getElementsByTagName(
→'body')[0]).appendChild(dsq);
    })();
</script>
<noscript>Please enable JavaScript to view the <a href="https://disqus.com/?ref_
→noscript" rel="nofollow">comments powered by Disqus.</a></noscript>
```

Comments using Django comments framework

Note: Please note that using Django's internal commenting is no longer recommended by Django community. Andablog uses it in the demo app to serve as an example of someone having their own custom comments system.

The comments count snippet is used to provide the necessary comment count. The comments snippet is for listing the comments below the entry.

The demo app has an example of overriding the snippets.

Sitemap Support

Andablog provides a andablog.sitemaps.EntrySitemap class that can be used within The Sitemap Framework.

The demo app has an example using the EntrySitemap.

RSS Feed Support

Andablog provides a djangoandablog.feeds.LatestEntriesFeed base class that can be sub-classed to provide a blog entries feed class to The Syndication Feed Framework.

The demo app has an example feed subclass.

Customizing the Author Display

Any Entry may have an Author, which is a foreignkey to the settings.auth_user_model Model. This auth.User by default or possibly a site's custom user model.

When displaying the author on templates, Andablog uses the andablog_tags.author_display tag to display the author and possibly link to a profile page:

- For Author display: The User model's get_short_name method is called. If not provided, the is used.

- For a hyperlink to an Author page: The User model's get_absolute_url method is called. If this method is absent or returns None/"" the author's display name is not hyperlinked.

Hint: If your site implements it's own comment or profile page system you may find the andablog_tags.author_display tag to be useful for the display of other users as well.

The demo app has an example custom user implementation.

Blog Entry Tags

The Entry model has a tags field provided by the django-taggit dependency. Out of the box this gives Andablog users the ability to add tags to an entry and manage them within the admin.

At the moment Andablog does not provide any template examples or tags that display them for you.

There is a (no longer maintained) django-taggit-templatetags project and some (maintained) offshoots to consider. They weren't up to date enough to package within Andablog.

1.1.3 Package Dependencies

- Python 3.5 or 2.7
- Django: Current and previous release
- six
- django-model-utils
- django-markupfield
- django-taggit
- Pillow

Features

This list will likely grow slowly. Priorities are Bug Fixes > Django Release Compatibility > Bad Jokes > Features.

- Blog administration through Django admin
- Markdown, RST or plain text support through django-markupfield
- Blog Entry tag management through django-taggit.
- Template block names are prefixed as to not conflict with the those used by the site.
- A URL hierarchy to include at /blog (or wherever)
- A Django sitemaps EntrySitemap class
- Preview content/image fields for entry for tighter control of how a blog entry looks in a listing w/fallback to truncation.
- A base class for an entries feed
- Utilizing a site-provided profile page as the author profile page
- Easy comment integration. Simply override a template snippet
- Support for custom User Models
- Django migrations
- Class based generic views that can be used directly
- A demo application.
- Django upgrade friendly: Most recently released major Django version and 1 back

Not Features

These features are right out. If you are looking for one of them, andablog may not be right for you.

- A User model. Andablog uses the settings.auth_user_model relation string for the author.

- Author Profile pages. These can be implemented by the site and linked to by andablog.

- Comments on blog entries. Though help is provided. In the form of a template snippet reference as well as a template tag that can be used for user display/linking.

- Constructing the author display name or URL. A provided User model must implement get_short_name for author display and get_absolute_url for author profile linking.

- Search. Since Andablog is only intended to be packaged with an existing site it would most likely become redundant.

- Support for 3 or more Django major releases. Sorry, if you want to proceed you will have to fork until your site catches up.

Trying out the demo site

4.1 Demo Site

4.1.1 No link?

Sorry, we don't yet have the demo hosted somewhere. To try out Andablog you have to pull down the source and run it locally.

4.1.2 Running Locally

The best way to test out the demo site is to set it up with all fixture data (so there is something to look at).

Using build scripts

If you have the build tools installed:

```
$ pynt create_venv
$ pynt rebuild_db
$ pynt runserver
```

Manually

1. Create and activate a virtualenv (somewhere)

2. Change directory to the django-andablog cloned dir

3. Install Requirements:

   ```
   $ pip install -r local_requirements.txt
   ```

4. Recreate the db and setup the database schema:

```
$ cd demo
$ python manage.py reset_db
$ python manage.py migrate
```

5. Load all fixtures

    Run this command for every 'fixtures' directory in the project:

    ```
    $ python manage.py loaddata someapp/fixtures/*.json
    ```

6. Run the server:

    ```
    $ python manage.py runserver
    ```

4.1.3 Pre-packaged Users

The demo fixtures include the following users. All users have 'secret' for a password.

Admins

- admin@example.com

Authors

- agent0014@example.com

Regular Users

- superman@example.com
- superwoman@example.com

Contributing to the project

Checkout the project page

5.1 Indices and tables

- genindex
- modindex
- search