Aldryn News Blog Documentation

Release 1.2.1

Divio AG

March 18, 2016

Contents

Aldryn News & Blog is an Aldryn-compatible news and weblog application for django CMS.

Content editors looking for documentation on how to use the editing interface should refer to our Using Aldryn News & Blog section.

Django developers who want to learn more about django CMS, as well as how to install, configure and customize it for their own projects should refer to the How-to guides and Reference sections.

Aldryn News & Blog is intended to serve as a model of good practice for development of django CMS and Aldryn applications.

CHAPTER 1

Documentation

1.1 How-to guides

These guides presuppose some familiarity with django CMS.

1.1.1 Installation

You can install Aldryn News & Blog either on Aldryn or by hand into your own project.

Aldryn Platform Users

To install the addon on Aldryn, all you need to do is follow this installation link on the Aldryn Marketplace and follow the instructions.

Manually you can:

1. Choose a site you want to install the add-on to from the dashboard.
2. Go to Apps > Install App
3. Click Install next to the News & Blog app.
4. Redeploy the site.

Manual Installation

Requirements

- This project requires **django CMS 3.0.12** or later.

PIP dependency

If you're installing into an existing django CMS project, you can run either:

```
pip install aldryn-newsblog
```

or:

```
pip install -e git+https://github.com/aldryn/aldryn-newsblog.git#egg=aldryn-newsblog
```

If you need to start a new project, we recommend that first you use the django CMS Installer to create it, and then install Aldryn News & Blog on top of that.

settings.py

In your project's `settings.py` make sure you have all of:

```
'aldryn_apphooks_config',
'aldryn_boilerplates',
'aldryn_categories',
'aldryn_common',
'aldryn_newsblog',
'aldryn_people',
'aldryn_reversion',
'aldryn_translation_tools',
'djangocms_text_ckeditor',
'easy_thumbnails',
'filer',
'parler',
'reversion',
'sortedm2m',
'taggit',
```

listed in `INSTALLED_APPS`, *after* `'cms'`.

Additional Configuration

Important: To get Aldryn News & Blog to work you need to add additional configurations:

1. Aldryn-Boilerplates You need set additional configurations to `settings.py` for Aldryn Boilerplates.

To use the old templates, set `ALDRYN_BOILERPLATE_NAME='legacy'`. To use https://github.com/aldryn/aldryn-boilerplate-bootstrap3 (recommended), set `ALDRYN_BOILERPLATE_NAME='bootstrap3'`.

2. Django-Filer Aldryn News & Blog requires the use of the optional "subject location" processor from Django Filer for Easy Thumbnails. This requires setting the `THUMBNAIL_PROCESSORS` tuple in your project's settings and explicitly omitting the default processor `scale_and_crop` and including the optional `scale_and_crop_with_subject_location` processor. For example:

```
THUMBNAIL_PROCESSORS = (
    'easy_thumbnails.processors.colorspace',
    'easy_thumbnails.processors.autocrop',
    # 'easy_thumbnails.processors.scale_and_crop',
    'filer.thumbnail_processors.scale_and_crop_with_subject_location',
    'easy_thumbnails.processors.filters',
    'easy_thumbnails.processors.background',
)
```

For more information on this optional processor, see the documentation for Django Filer.

Migrations

Now run `python manage.py syncdb` if you have not already done so, followed by `python manage.py migrate` to prepare the database for the new applications.

Note: Aldryn News & Blog supports both South and Django 1.7 migrations.

Server

To finish the setup, you need to create a page, change to the *Advanced Settings* and choose *NewsBlog* within the *Application* drop-down.

You also need to set the *Application configurations* and **publish the changes**.

Finally you just need to **restart your local development server** and you are ready to go.

This process is described in more depth within Basic Usage.

1.1.2 Upgrading

The CHANGELOG is maintained and updated within the repository.

Upgrade from 0.5.0

Note: If you're upgrading from a version earlier than 0.5.0.

In this version 0.5.0, we're deprecating all of the static placeholders and instead making them PlaceholderFields on the app_config object. This means that you'll be able to have content that is different in each instance of the app, which was originally intended.

Because some may have already used these static placeholders, there will be a (very) short deprecation cycle. 0.5.0 will introduce the new PlaceholderFields whilst leaving the existing static placeholders intact. This will allow developers and content managers to move plugins from the old to the new.

Version 0.6.0 will remove the old static placeholders to avoid any further confusion.

Also note: The article's PlaceholderField has also had its visible name updated. The old name will continue to be displayed in structure mode until the article is saved. Similarly, the new app_config-based PlaceholderFields will not actually appear in structure mode until the app_config is saved again.

1.1.3 Basic Usage

Aldryn News & Blog works the way that many django-CMS-compatible applications to. It expects you to create a new page for it in django CMS, and then attach it to that page with an Apphook.

Getting started

1. if this is a new project, change the default *example.com* `Site` in the Admin to whatever is appropriate for your setup (typically, *localhost:8000*)
2. in Admin > Aldryn_Newsblog, create a new `Apphook config` with the value *aldryn_newsblog*

3. create a new django CMS page; this page will be associated with the Aldryn News & Blog application
4. open the new page's `Advanced settings`
5. from the `Application` choices menu select *NewsBlog*
6. save the page
7. restart the runserver (necessary because of the new Apphook)

Now you have a new page to which the Aldryn NewsBlog will publish content.

Let's create a new weblog article, at Admin > Aldryn_NewsBlog. Fill in the fields as appropriate - most are self-explanatory - and **Save**.

The page you created a moment ago should now list your new article.

1.2 Reference

1.2.1 Settings

The flag `ALDRYN_NEWSBLOG_SEARCH` can be set to `False` in settings if indexing should be globally disabled for Aldryn News & Blog. When this is `False`, it overrides the setting in the application configuration on each apphook.

If Aldryn Search, Haystack, et al, are not installed, this setting does nothing.

The flag `ALDRYN_NEWSBLOG_UPDATE_SEARCH_DATA_ON_SAVE`, when set to True (default value), updates the article's search_data field whenever the article is saved or a plugin is saved on the article's content placeholder. Set to false to disable this feature.

1.2.2 Management Commands

The management command: `rebuild_article_search_data` can be used to update the search_data in all articles for searching. It can accept a switch `--language` or the short-hand `-l` to specify the translations to process. If this switch is not provided, all translations are indexed by default.

1.2.3 Plugins

Related Articles Plugin

The Related Articles plugin is only appropriate for use only on the article detail view. If the plugin in placed on any other page, it will render an empty `<div></div>`.

1.3 Using Aldryn News & Blog

The documentation in these two sections focuses on the basics of content creation and editing using Aldryn News & Blog. It's suitable for non-technical and technical audiences alike.

1.4 Development & community

Aldryn News & Blog is an open-source project.

You don't need to be an expert developer to make a valuable contribution - all you need is a little knowledge, and a willingness to follow the contribution guidelines.

1.4.1 Divio AG

Aldryn News & Blog is developed by Divio AG and released under a BSD licence.

Aldryn News & Blog is compatible with Divio's Aldryn cloud-based django CMS hosting platform, and therefore with any standard django CMS installation. The additional requirements of an Aldryn application do not preclude its use with any other django CMS deployment.

Divio is committed to Aldryn News & Blog as a high-quality application that helps set standards for others in the Aldryn/django CMS ecosystem, and as a healthy open source project.

Divio maintains overall control of the Aldryn News & Blog repository.

1.4.2 Standards & policies

Aldryn News & Blog is a django CMS application, and shares much of django CMS's standards and policies.

These include:

- guidelines and policies for contributing to the project, including standards for code and documentation
- standards for managing the project's development
- a code of conduct for community activity

Please familiarise yourself with this documentation if you'd like to contribute to Aldryn News & Blog.

1.4.3 Running tests

Aldryn News & Blog uses django CMS Helper to run its test suite.

Backend Tests

To run the tests, in the aldryn-newsblog directory:

```
virtualenv env  # create a virtual environment
source env/bin/activate  # activate it
python setup.py install  # install the package requirements
pip install -r test_requirements/django-1.7.txt  # install the test requirements
python test_settings.py  # run the tests
```

You can run the tests against a different version of Django by using the appropriate value in `django-x.x.txt` when installing the test requirements.

Frontend Tests

Follow the instructions in the aldryn-boilerplate-bootstrap3 documentation and setup the environment through the *Backend Tests* section.

Instead of using `python test_settings.py` described above, you need to excecute `python test_settings.py server` to get a running local server. You can open the development server locally through `http://127.0.0.1:8000/`. The database is added within the root of this project `local.sqlite`. You might want to delete the database from time to time to start with a fresh installation. Don't forget to restart the server if you do so.

1.4.4 Documentation

You can run the documentation locally for testing:

1. navigate to the documentation `cd /docs`
2. run `make install` to install requirements
3. run `make run` to run the server

Now you can open **http://localhost:8000** on your favourite browser and start changing the rst files within `docs/`.