Aldryn FAQ Documentation

Release 2.1.2

Divio AG

May 07, 2018

Contents

Aldryn FAQ is an Aldryn-compatible simple Frequently Asked Questions (FAQ) applicatio for django CMS.

It allows you to present categorized lists of frequently asked questions and their answers.

Content editors looking for documentation on how to use the editing interface should refer to our Using Aldryn FAQ 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.

Documentation

How-to guides

These guides presuppose some familiarity with django CMS.

Installation

You can install Aldryn FAQ 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 FAQ app.

4. Redeploy the site.

Manual Installation

Requirements

- This project requires **django CMS 3.4** or later.

PIP dependency

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

```
pip install aldryn-faq
```

or:

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

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 FAQ on top of that.

settings.py

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

```
'adminsortable2',
'aldryn_boilerplates',
'aldryn_translation_tools',
'djangocms_text_ckeditor',
'parler',
'sortedm2m',
'aldryn_faq',
'taggit',
```

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

Additional Configuration

Important: To get Aldryn FAQ 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-Parler If you plan to use translations, configure django-parler.

Be sure to add `PARLER_LANGUAGES` to your settings with the appropriate configuration for your project. Example:

```
PARLER_LANGUAGES = {
    1: (
        {'code': 'en',},
        {'code': 'fr',},
        {'code': 'de',},
    ),
    'default': {
        'fallback': 'en',              # defaults to PARLER_DEFAULT_LANGUAGE_CODE
        'hide_untranslated': False,   # the default; let .active_translations() return fallbacks too.
    }
}
```

Migrations

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

Now run `python manage.py migrate aldryn_faq`.

Server

To finish the setup, you need to create a page, change to the *Advanced Settings* and choose *FAQ* 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.

Upgrading

The CHANGELOG is maintained and updated within the repository.

Basic Usage

Aldryn FAQ 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_Faq, create a new `Apphook config` with the value *aldryn_faq*

3. create a new django CMS page; this page will be associated with the Aldryn FAQ application

4. open the new page's `Advanced settings`

5. from the `Application` choices menu select *FAQ*

6. save the page

7. restart the runserver (necessary because of the new Apphook)

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

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

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

Reference

Features

- Full internationalization; questions and their answers can be managed in multiple lanauges.

- Support for "spaces" (apphook_config-able) so that multiple, independent instances of this app can be attached to multiple pages in the same project, if necessary.

- Support for Django Reversion which allows content managers to roll-back to previous edits should this be necessary.

django-admin-sortable2

This project uses django-admin-sortable2 version 0.5.0 or later. Installing an earlier version of django-admin-sortable2 will cause migrations to fail due to a package name clash with django-admin-sortable.

Using Aldryn FAQ

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

Development & community

Aldryn FAQ 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.

Divio AG

Aldryn FAQ is developed by Divio AG and released under a BSD licence.

Aldryn FAQ 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 FAQ 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 FAQ repository.

Standards & policies

Aldryn FAQ 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 FAQ.

Running tests

Aldryn FAQ uses django CMS Helper to run its test suite.

There's more than one way to do this, but here's one to help you get started:

```
# create a virtual environment
virtualenv test-aldryn-faq

# activate it
cd test-aldryn-faq/
source bin/activate

# get Aldryn FAQ from GitHub
git clone git@github.com:aldryn/aldryn-faq.git

# downgrade pip to a version < 6
pip install -U 'pip<6'

# install the dependencies for testing
pip install -Ur aldryn-faq/test_requirements.txt

# run the test suite
# note that you must be in the aldryn-faq directory when you do this,
# otherwise you'll get "Template not found" errors
cd aldryn-faq
./test
```

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/`.