Aldryn Events Documentation

Release 1.0

Divio AG

January 19, 2018

Contents

Aldryn Events is an Aldryn-compatible events application for for django CMS.

Aldryn Events allows you to publish and manage a calendar of events - along with their start and end times, location and so on - on your django CMS website. It's easy to use and flexible, and includes advanced features such as registration.

Aldryn Events is open-source software.

CHAPTER 1

Introduction

Installation

Installing packages

We'll assume you have a django CMS (version 3.x) project up and running.

If you need to set up a new django CMS project, follow the instructions in the django CMS tutorial.

Then run either:

```
pip install aldryn-events
```

or to install from the latest source tree:

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

Edit `settings.py`

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

```
'aldryn_apphooks_config',
'aldryn_boilerplates',
'aldryn_translation_tools',
'aldryn_common',
'aldryn_events',
'appconf',
'bootstrap3',
'django_tablib',
'djangocms_text_ckeditor',
'easy_thumbnails',
'extended_choices',
'filer',
'parler',
'sortedm2m',
'standard_form',
```

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

Aldryn Boilerplates

This application uses (and will install) Aldryn Boilerplates, which requires some basic configuration to get you started.

Note: If you are using Django 1.8 please note the configuration instructions for Aldryn Boilerplates.

Edit your settings so that they conform to:

```
TEMPLATE_CONTEXT_PROCESSORS = [
    ...
    'aldryn_boilerplates.context_processors.boilerplate',
    ...
]

STATICFILES_FINDERS = [
    'django.contrib.staticfiles.finders.FileSystemFinder',
    # important - place immediately before AppDirectoriesFinder
    'aldryn_boilerplates.staticfile_finders.AppDirectoriesFinder',
    'django.contrib.staticfiles.finders.AppDirectoriesFinder',
]

TEMPLATE_LOADERS = [
    'django.template.loaders.filesystem.Loader',
    # important! place right before django.template.loaders.app_directories.Loader
    'aldryn_boilerplates.template_loaders.AppDirectoriesLoader',
    'django.template.loaders.app_directories.Loader',
]
```

Now set the name of the boilerplate you'll use in your project:

```
ALDRYN_BOILERPLATE_NAME = 'bootstrap3'
```

Note: Note that Aldryn Events doesn't use the the traditional Django `/templates` and `/static directories`. Instead, it employs Aldryn Boilerplates, which makes it possible to to support multiple different frontend schemes ('Boilerplates')and switch between them without the need for project-by-project file overwriting.

Aldryn Events's templates and static files will be found in named directories in the `/boilerplates` directory.

Filer

Aldryn Events also depends on Filer, be sure to follow Filer's installation instructions. To get up and running quickly, make sure you adapt your settings to include the `filer.thumbnail_processors.scale_and_crop_with_subject_location` thumbnail processor:

```
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',
)
```

Prepare the database and run

Now run `python manage.py migrate` to prepare the database for the new application, then `python manage.py runserver`.

For Aldryn users

On the Aldryn platform, the Addon is available from the Marketplace.

You can also install Aldryn Events into any existing Aldryn project.

Basic usage

Create a django CMS page to hook the Aldryn Events application into. In its *Advanced settings*, set its `Application` to *Events*, and use the provided, default *Events / aldryn_events* in `Application configurations` before saving.

Note: If you are upgrading from version 0.7.x or lower, please make sure to visit the page *Advanced settings* (as well as every Aldryn Events plugin configuration), select application configuration and save it.

The page is now a landing page for events on the site and will show a calendar (empty), which will be populated as you add events to the system.

The behaviour of the Events system should be largely self-explanatory, but the tutorial for users will guide you through some basic steps if necessary.

CHAPTER 2

Release notes & upgrade information

The CHANGELOG is maintained and updated within the repository.

CHAPTER 3

Using Aldryn Events

Note: This part of the guide assumes that django CMS has been appropriately installed and configured, including Aldryn Events, and that an Events page has been set up.

Add an event

Visit the Events page. You should see that the django CMS toolbar now contains a new item, *Events*. Select *Add Event...* from this menu.

Provide some basic details, such as:

- the `Short description` is a brief summary of the Event, that will be used in lists of Events
- an event must have a `Start date`, but the other date/time fields are optional
- for the `Location`, enter as complete address as possible - Aldryn Events will pass this on to Google Maps to display a map, so it needs to be unambiguous and accurate

and **Save** your event.

It now exists in the database and will be listed on the Events page. Notice that the calendar also indicates that something's on.

You can use the standard django CMS placeholder interface to add more content to your event.

Advanced settings

An event's *Advanced* settings include:

- precise `Location latitude`/`Location longitude` (this overrides the geolocation based on the `Location` field)
- Event registration, that allows you to collect data from prospective attendees - their replies are stored in the database (see *Aldryn Events* > *Registrations* in the Admin.)
- Event co-ordinators, who will receive the registration messages
- an external registration link - for example, if you're using a service such as Tito

Plugins

The Events page is the easy way in to events on the system, but Aldryn Events also includes a number of plugins that can be inserted into any django CMS page - indeed, into any content - to deliver information about events.

For example, if you have a news article announcing a series of seminars, you can drop an Events plugin into that page to insert an automatically-updated list of items.

Calendar

A month view, similar to the one that appears on the main Events page.

List

Choose the events you wish to have displayed.

Upcoming or past events

An automatic list of events.

CHAPTER 4

Development & community

Aldryn Events 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 Events was created by Divio AG and is released under a BSD licence.

Aldryn Events 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 Events 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 Events repository.

Standards & policies

Aldryn Events 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 Events.

Running tests

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

Backend Tests

To run the tests, in the aldryn-events 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_17.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_xx.txt` when installing the test requirements.

Frontend Tests

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

Instead of using `python test_settings.py` described above, you need to execute `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.