django-magazine Documentation

Release 0.1.5

Dominic Rodger

Jul 09, 2017

Contents

django-magazine is a pluggable Django app for managing a simple magazine.

Magazines consist of Issues, each of which contains one or more Articles, which are by one or more Authors.

CHAPTER 1

Get the code

The source is available on Github.

CHAPTER 2

Contents:

Installation

Note: django-magazine is not yet available on PyPI - there's some cleanup I need to do before I can make it available there.

Before you start

You'll need Python and virtualenv if you don't already have them. Using a virtual environment will make the installation easier, and will help to avoid clutter in your system-wide libraries. You will also need Git to clone the repository.

Trying it out

django-magazine ships with an example project so you can see what it does quickly. Installation is as follows:

```
git clone http://github.com/dominicrodger/django-magazine.git
cd django-magazine
virtualenv magazine
source magazine/bin/activate
pip install -r requirements.txt
pip install Django
cd magazine/example_project
python manage.py syncdb --noinput
python manage.py loaddata sample_magazine_data.json
python manage.py test magazine
python manage.py runserver
```

This'll add a superuser with the username `admin` and the password `admin`, so you can play around with the admin site, which will be at http://127.0.0.1:8000/admin/. It'll also create a sample issue with a couple of articles.

Installing django-magazine

django-magazine is not yet available on PyPI, but in the mean time you can install from GitHub like this:

```
pip install git+git://github.com/dominicrodger/django-magazine.git
```

Then add `magazine` and `tinymce` to your `INSTALLED_APPS` setting, and update your database using `syncdb` or `migrate` (if you're using South).

Models

Note: django-magazine also has another model which the documentation does not refer to, since it will be removed soon - BookReview. This model was relevant to one particular use case of django-magazine, but is unlikely to be useful to most people.

There are 3 main models in django-magazine: Issue, Article and Author. An Issue is made up of Articles, each of which by at least one Author.

Issue

Issues have three main properties: number, date, and whether or not it is published.

Issue Number These are intended to be used as sequential issue numbers, and must be unique.

Date This is the publication date of the issue - note that an issue is only considered published if this is on or before the current date, and the published flag is `True`.

Published Used for creating draft issues - you can safely mark an issue with a date in the future as published - it will only be publically accessible once the publication date is reached.

Embargoing Issues

Sometimes it's useful to just show teasers of content - perhaps for very recent content, or for old content, perhaps to encourage users to subscribe to a dead-tree version. At present, this functionality isn't very fleshed out (it just shows teaser content to everyone other than site staff), in future, I'm hoping this will take into account the current user, to allow showing full content if you're logged in, for example.

There are two settings relevant to this feature:

MAGAZINE_IS_EMBARGOED_FUNCTION Set to a function which takes an `Issue` object and returns `True` if that issue is embargoed.

MAGAZINE_EMBARGO_TIME_IN_MONTHS If `MAGAZINE_IS_EMBARGOED_FUNCTION` is not set, then this will determine how many months old an issue has to be before it is no longer embargoed.

If an issue is embargoed, a teaser will be shown. The teaser is either the article's description, or the first 50 words of the article.

Article

Articles are the content of magazines. Each article belongs to an issue, and is by one or more authors.

Articles are ordered within an issue, based on the value of `order_in_issue` (articles with lower values are shown first).

Note: Article text will be cleaned up automatically with bleach, which strips out bad HTML that tends to be added by programs like Microsoft Word. The only allowed tags are those set by `bleach.ALLOWED_TAGS`, `<p>`, and `<h[1-5]>`.

Author

Authors have names, and a quick biography.

Sometimes, you don't want users to be able to see a list of articles by a particular author (e.g. when importing articles where the author is unknown, you might want to attribute the article to "Unknown", so that something shows up where the list of authors would normally be). To do this, just uncheck the `indexable` value of the author.

Templates

django-magazine is designed to be as easy as possible to style with CSS, or by overriding templates. All templates live in `magazine/`.

The base template is `magazine/magazine_base.html` - a default ships with the app, but you probably want to override it to apply your styling. All other templates extend `magazine/magazine_base.html`.

To see what other templates are overridable, take a look at the templates directory.