Django Website Showroom Documentation

Release 0.4

Holger Drewes

Apr 27, 2017

Contents

Python/Django project to easily create showroom websites presenting different websites around a certain topic, originally used for http://opendata-tools.org (engl.) and http://opendata-showroom.org (german).

CHAPTER 1

Features

- Administrable via Django admin (Categories, Websites)
- Screenshots with automatic resize functionality
- Customizable category colors
- Creation of different language editions
- Edition-dependent custom static page texts (like title, subtitle, footer,...)
- Edition-dependent category and website selection
- No language dependent static code
- Responsive, mobile-friendly design

CHAPTER 2

Contents

Installation

Requirements

- Python 3.4+
- Django 1.11
- Hackstack 2.6 (Search module for Django)
- Whoosh 2.7 (Full-text search enginge used for django-haystack)
- Pillow

Installation with pip

Create a `virtualenv` environment, activate it and install the package with all the dependencies via:

```
pip install django-website-showroom
```

Manual Installation

- Clone source repository from GitHub
- Create a virtual environment with `virtualenv`
- Install the requirements with `pip install -r requirements.txt`
- Manually link the `website_showroom` folder to the `site-packages` folder of your environment

Setup

Create a separate `Django` project:

```
django-admin(.py) startproject my_showroom_project
```

Add `haystack` and the `website_showroom` app to `INSTALLED_APPS` in `settings.py`:

```
INSTALLED_APPS = (
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'haystack',
    'website_showroom',
)
```

The following settings are necessary for `Haystack` (search) to work:

```
# Haystack settings
HAYSTACK_CONNECTIONS = {
    'default': {
        'ENGINE': 'haystack.backends.whoosh_backend.WhooshEngine',
        'PATH': os.path.join(BASE_DIR, 'search_index'),
    },
}
HAYSTACK_SIGNAL_PROCESSOR = 'haystack.signals.RealtimeSignalProcessor'
```

Add the import for the Showroom urls to your `urls.py` file and add a + to the Django admin urls assignment to the `urlpatterns` list:

```
from website_showroom.urls import urlpatterns

# Before: urlpatterns = [...]
# After: urlpatterns += [...]
```

Run Django migration command to create the DB tables and create an admin user:

```
python manage.py migrate
python manage.py createsuperuser
```

Now run the Django server command:

```
python manage runserver
```

You should be able to enter the Django admin with your user credentials at http://127.0.0.1:8000/admin/.

Example Project

There is an example project where you can see a showroom in action!

- Go to the `example_project` folder of the showroom lib installation
- Create the DB with `python manage.py migrate` and load the initial example data with `python manage.py loaddata example_project.json`

- Rebuild the search index with `python manage.py rebuild_index`
- Run a Django server with `python manage runserver`
- Admin credentials are "admin/admin"

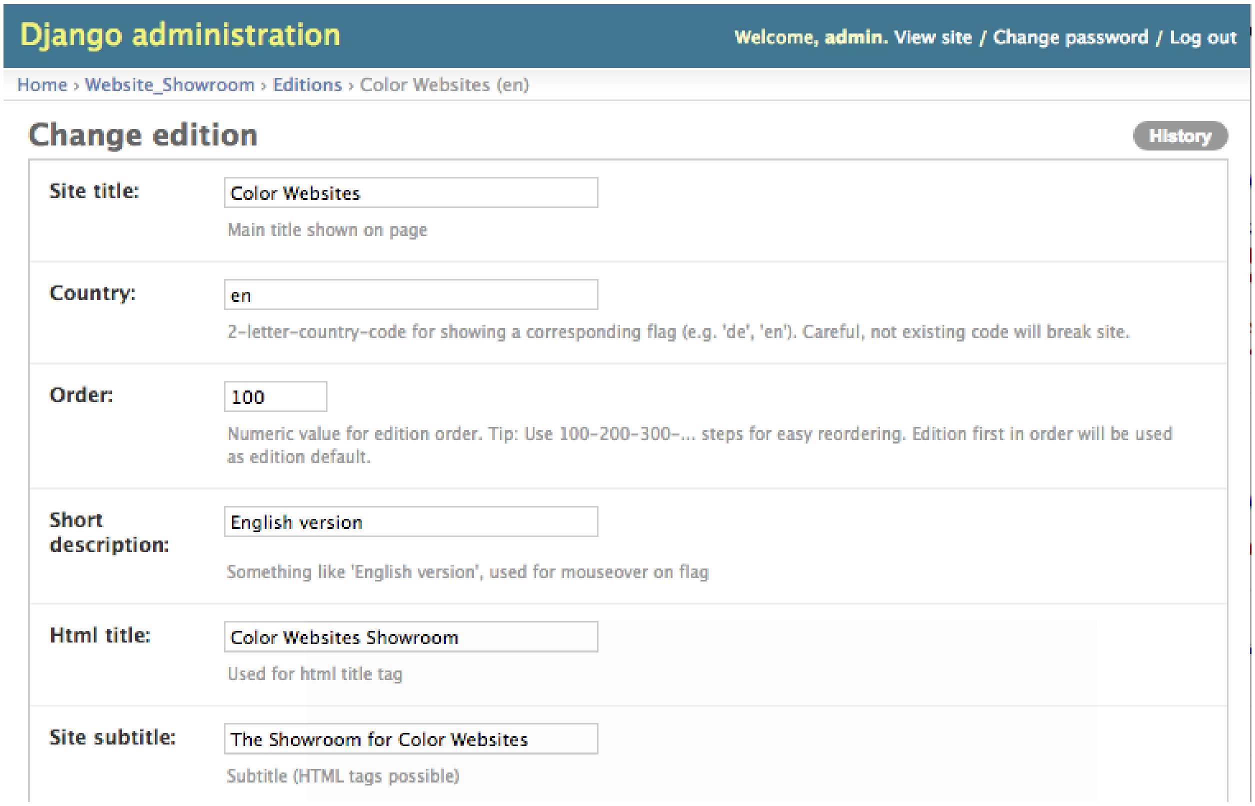
Create your Showroom

After you have installed and configured the library you can actually start to create your showroom of websites.

Add an Edition

A website showroom consists of one or more `editions`, representing the version of the showroom for a specific country/language.

You can add en edition of the site in the `Django admin`. For creating an edition, open an edition form and follow the instructions close to the form fields.


Django administration
Welcome, admin. View site / Change password / Log out
Home › Website_Showroom › Editions › Color Websites (en)
History

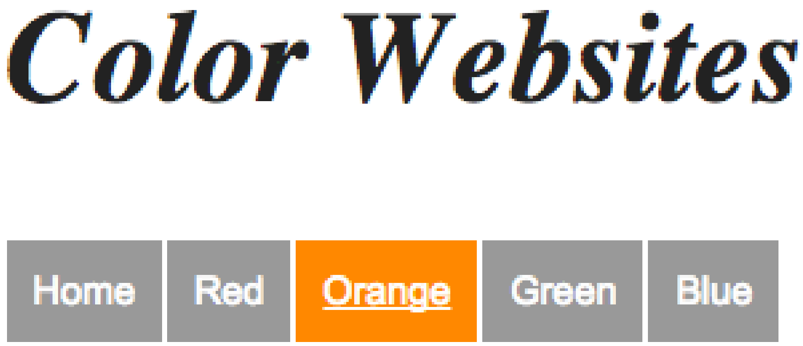
Change edition

Site title:	Color Websites Main title shown on page
Country:	en 2-letter-country-code for showing a corresponding flag (e.g. 'de', 'en'). Careful, not existing code will break site.
Order:	100 Numeric value for edition order. Tip: Use 100-200-300-... steps for easy reordering. Edition first in order will be used as edition default.
Short description:	English version Something like 'English version', used for mouseover on flag
Html title:	Color Websites Showroom Used for html title tag
Site subtitle:	The Showroom for Color Websites Subtitle (HTML tags possible)

After adding an edition entry, you should for the first time be able to run the front-end website and see the basic layout of the site.

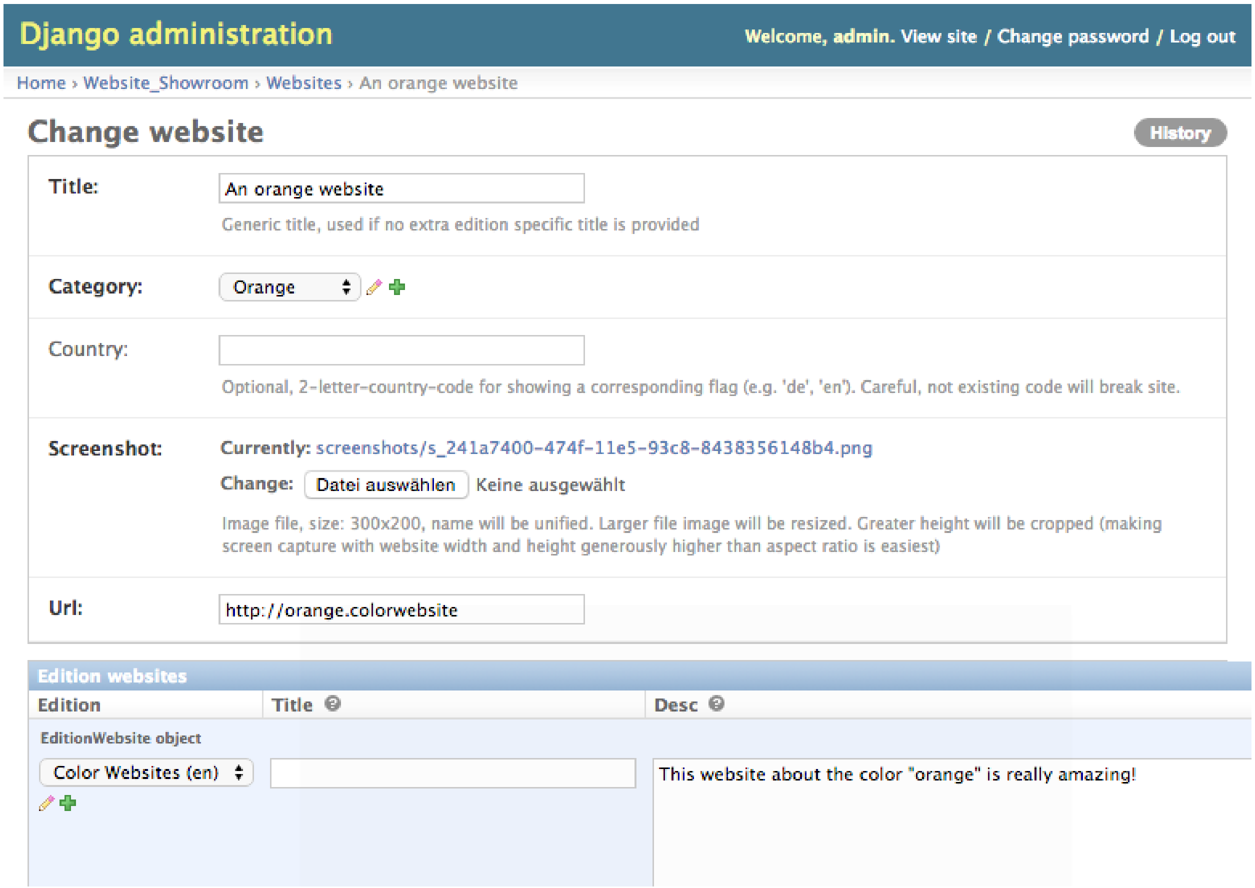
Create Categories

Categories will show as the main navigation of the showroom. You have to explicitly name the edition(s) a category should be displayed/associated. This allows for having (slightly) different categories for different editions.



Add your Websites

Now to the actual fun: look for websites about the topic you would want to make the website showroom about and add them to your site via the `Django admin`.


Django administration
Welcome, admin. View site / Change password / Log out
Home › Website_Showroom › Websites › An orange website
History

Change website

Title: An orange website
Generic title, used if no extra edition specific title is provided

Category: Orange

Country:
Optional, 2-letter-country-code for showing a corresponding flag (e.g. 'de', 'en'). Careful, not existing code will break site.

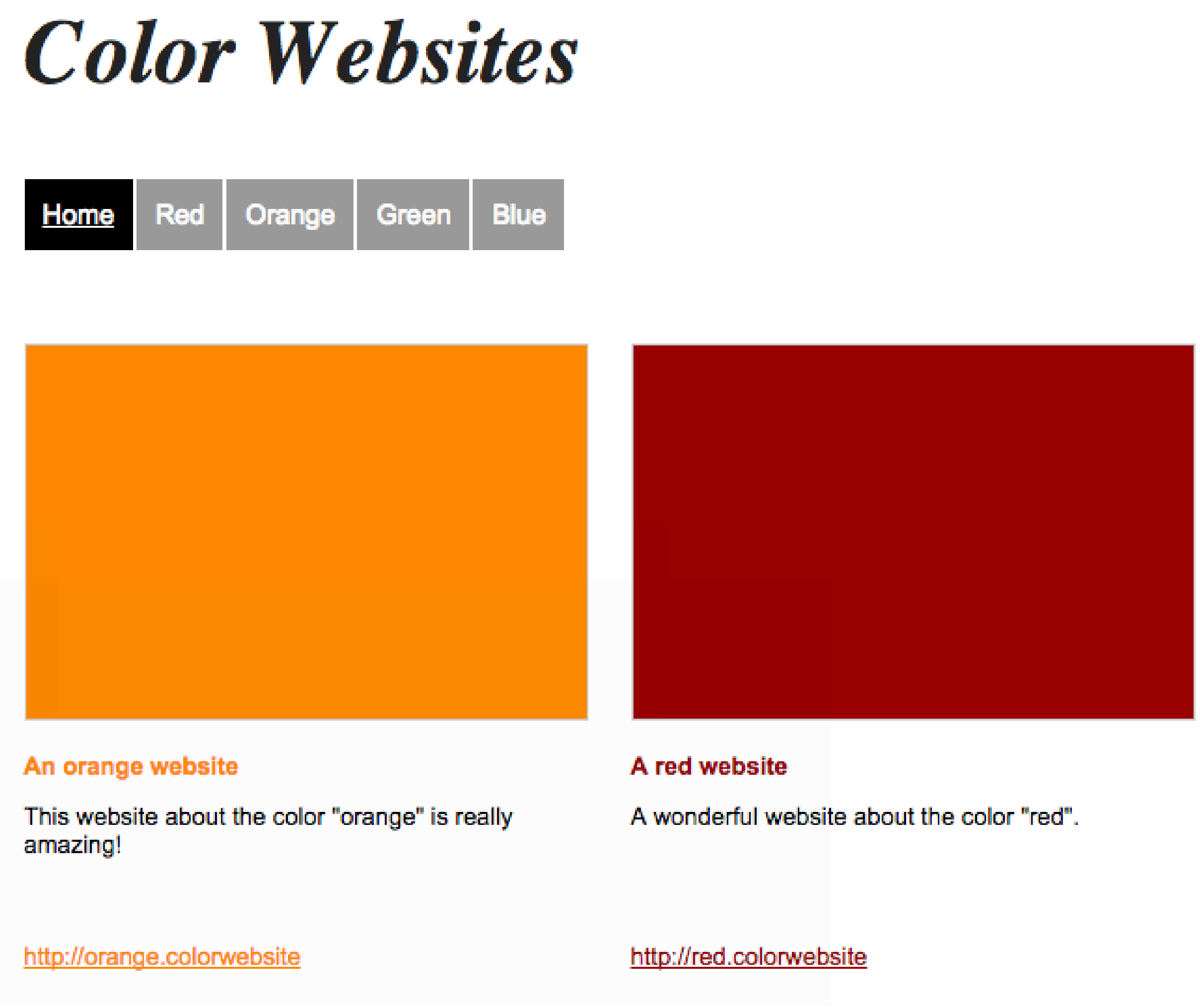
Screenshot: Currently: screenshots/s_241a7400-474f-11e5-93c8-8438356148b4.png
Change: Datei auswählen Keine ausgewählt
Image file, size: 300x200, name will be unified. Larger file image will be resized. Greater height will be cropped (making screen capture with website width and height generously higher than aspect ratio is easiest)

Url: http://orange.colorwebsite

Edition websites

Edition	Title	Desc
EditionWebsite object		
Color Websites (en)		This website about the color "orange" is really amazing!

You can then see your websites in the editions you added them to (language/country editions can be switched via the flags in the top-right corner of the showroom website).



Color Websites

An orange website

This website about the color "orange" is really amazing!

http://orange.colorwebsite

A red website

A wonderful website about the color "red".

http://red.colorwebsite

And that’s it.

Now add your own stuff! :-)

Development

Release Notes

Changes in version 0.4.0 (2017-04-27)

- `Python 3.4+` support, `Python 2.7` support dropped
- Updated `Django` requirement version from 1.8 to 1.11
- Updated `Haystack` dependency to 2.6, `Whoosh` dependency to 2.7

Changes in version 0.3.1 (2015-08-25)

- Added more entries to the example project (see: *Example Project*)
- Switched back to 3-column layout for large screens
- Layout improvements

- Removed not-used library files
- Fixed bug when using same category URL name for different editions

Changes in version 0.3 (2015-08-24)

- Responsive design
- Bug fixes

Changes in version 0.2 (2015-08-20)

- New organized development structure with new separate docs (this one), branch-based development
- Made `Haystack` work again, fixed requirements to `django-haystack==2.0.0` and `Whoosh==2.4.1` (new setting HAYSTACK_SIGNAL_PROCESSOR = 'haystack.signals.RealtimeSignalProcessor' in `settings.py` necessary)
- Replaced `PIL` requirement with `Pillow`
- Support for `Django 1.8` (older versions dropped), coming from `1.4` following adoptions are necessary:
  - `ALLOWED_HOSTS` has to be added to `settings.py` of Django project
  - `python manage.py migrate` has to be run to apply/recognize the new `Django` migrations
- Replaced `South` migrations with re-generated `Django` internal migrations
- Setup instructions in docs
- New `example_project` Django project with basic Color Website Showroom example
- Section in docs describing how to create a showroom website
- Fixed some bugs

Changes in version 0.1 (A long time ago...)

- Initial version, just existing in `master` branch, no dedicated `tags` or `pip` releases yet

CHAPTER 3

Indices and tables

- genindex
- modindex
- search