
django Organice Documentation

Release 0.1

Peter Bittner <django@bittner.it>

February 28, 2014

Contents:

django Organice

All-in-one collaboration solution. For non-profit organizations, sports clubs, small to medium-sized businesses. Nice, very nice, <http://organice.io>

django Organice is a nice way to run a collaboration platform, an intranet, and websites for your business. It's a compilation of the best Django packages, preconfigured for getting you started quickly. Being powered by the well-known [Django Web framework](#) it's a safe choice for your IT investment providing an easily extensible architecture.

Primary goal of the software is to provide an absolutely intuitive, consistent user experience. Because in fast-paced, flexible, probably small-margin businesses you don't have the time to invest in training your staff. We build on promising, reliable Django components, and rather invest in those projects than build source code here. We expect to continually tune the compilation to keep the source footprint small, keeping the project lean and responsive.

See <http://organice.io> for more information.

1.1 Features

Major components of django Organice are:

- Django Web framework
- full-featured content management (django CMS)
- full-featured versatile blog (Zinnia, cmsplugin)
- events with event calendar (via Zinnia)
- flexible contact pages (cmsplugin)
- link collection pages
- full-featured newsletter (Emencia)
- user profiles (userena)
- multi-language support
- multiple websites support
- database support for PostgreSQL, MySQL, and more

Features on the roadmap include:

- social login (Python Social Auth)
- group spaces with automatic mailing list (`group_name@yoursite.tld`)
- issue list / task management

- documentation area / wiki (probably by use of CMS functionality)
- discussion forum / Q&A board (Askbot, Misago)
- content download for magazine generation
- unobtrusive, integrated, transparent document and asset management
- DropBox, OneDrive, Google Drive, iCloud, etc. integration
- fully integrated search (Haystack)
- web analytics respecting your privacy (Piwik)
- live chat or chat integration

1.2 Who is using django Organice?

Examples of websites running django Organice:

- Kreuzlingen Rudolf Steiner school

1.3 Download and Contributions

Official repositories: (kept in sync)

1. Bitbucket: <https://bitbucket.org/bittner/django-organice>
2. GitHub: <https://github.com/bittner/django-organice>

1.4 Getting Help

- Documentation is available at <http://docs.organice.io>
- Questions? Please use [StackOverflow](#). Tag your questions with `django-organice`.
- Found a bug? Please use either the [Bitbucket](#) or [GitHub](#) issue tracker (you choose)

Installation

This document assumes you are familiar with basic Python and Django development and their [tools](#). If not, please read up on [pip](#), [virtualenv](#), and [virtualenvwrapper](#) first. A basic understanding is sufficient.

2.1 Requirements

- Python 2.6 or higher

All other dependencies are resolved by the django Organice installer.

2.1.1 Recommended for installation

- pip
- virtualenv
- virtualenvwrapper

2.2 Installing django Organice

1. We recommend preparing a virtual environment for running django Organice:

```
$ mkvirtualenv example
$ workon example
```

The prompt will change to something like `(example) ~$` to reflect that your new virtual environment is active.

2. The easiest way is using `pip` for installation:

```
$ pip install django-organice
```

This will pull the latest django Organice package from the Internet and install all dependencies automatically.

If you're a developer you may want to run django Organice with the latest sources: (don't do this as a user)

```
$ git clone git@github.com:bittner/django-organice.git
$ cd django-organice
$ python setup.py install
```

or, alternatively, using pip:

```
$ pip install git+https://github.com/bittner/django-organice.git#egg=django-organice
```

3. Install the adapter suitable for your database (PostgreSQL `psycopg2`, MySQL `MySQL-python`, Oracle `cx_Oracle`, etc.), e.g.

```
$ pip install psycopg2
```

The Django project recommends PostgreSQL.

NOTE: You can skip this step if you decide to use SQLite, e.g. for evaluation purposes.

4. Run the Organice setup command to create your new project: (e.g. *example*)

```
$ organice-setup example
```

5. Edit your settings in `example/settings/common.py`, `example/settings/develop.py`, etc. See the [Django documentation](#) on settings if you're not familiar with it. The `develop` settings are used by your project by default (local development), `common` is included in all profiles.

6. Initialize your database and start rocking:

```
$ python manage.py syncdb --migrate
$ python manage.py runserver
```

You can now point your browser to <http://127.0.0.1:8000/> and start developing your project locally.

NOTE: If you're planning to create your content locally make sure you use the same database engine for local development and production. Your plan of moving the whole database content from development to production will give you major headaches otherwise. And, use SQLite for evaluating only!

2.3 Initial Configuration

1. Follow the instructions given to you by the django Organice installer `organice-setup` after setup has completed. You have to adapt some values in your project settings!
2. If you want your site to use a language other than English, or you want to use several languages: Adapt the values of `LANGUAGE_CODE` and `LANGUAGES`, and set `USE_I18N = True` in your project settings.
3. Add your first pages, blog posts, and newsletter data:
 - Add some pages and navigation in the Django administration at `Cms > Pages`, and publish your changes.
 - Surf your new website, and fill your new pages with content using the front-end editing feature.
 - Surf to `/blog/` on your website, and start adding Blog posts.
 - Add a user in the Django administration at `Newsletter > Contacts`.
 - Add `localhost` (or appropriate server) to `Newsletter > SMTP servers`.
 - To allow subscribing from the website (from `/newsletter/subscribe`) add a list to `Newsletter > Mailing lists`.
 - Finally, add your first newsletter to `Newsletter > Newsletters`.

- For adding templates to Emencia Newsletter please consult the related section in the [TinyMCE 3.x documentation](#).
4. For sending newsletters to work you must configure a cronjob polling on `python manage.py send_newsletter` every half an hour. If that was just Greek to you go ask your server admin for help. She knows!

2.4 Deployment to Production

During the installation `organice-setup` prepared 3 different environments that help you with deployment:

```
example
-- settings
|  -- __init__.py
|  -- common.py
|  -- develop.py
|  -- staging.py
|  -- production.py
```

This modularized setup is described in Solution 2 of Tommy Jarnac's blog on [Django settings best practices](#)¹. The `develop` settings are active by default (for local development), `common` is included by all profiles.

For deployment to environments other than `develop` the settings module location must be overridden by setting the Django environment variable `DJANGO_SETTINGS_MODULE`. For example, if you use Apache as your Django web server adapt your Apache configuration file for `example` with:

```
SetEnv DJANGO_SETTINGS_MODULE example.settings.production
```

NOTE: To test different settings locally you can start the Django webserver with the `--settings` option:

```
$ python manage.py runserver --settings=example.settings.staging
```

¹ David Cramer from DISQUS has described a similar solution at <http://justcramer.com/2011/01/13/settings-in-django/>

django Organice is composed of the following main components:

1. Content Management (Cms)
2. Blog (Zinnia)
3. Newsletter

3.1 Content Management

Editing your website is not much different from surfing the web. When you're logged in to your website you will have the django CMS toolbar on top of the page. Slide the "Edit mode" button to the "ON" state, and some elements of the page you're currently on start having additional toolbars to add and edit content in those areas.

You can safely modify content on the page without the website to change. Only you can see the changes, and only as long as you are in Edit mode. When you're happy with the changes press the blue "Publish" button on the django CMS toolbar to make your version visible to the world.

For more technical tasks like creating a navigation structure, adding pages, etc. the django CMS toolbar redirects you to the Django administration interface. Want to have an upfront first impression before using it? Check out the (slightly outdated) [django CMS video on frontend-editing](#).

3.2 Blog

Writing and publishing interesting articles now and then is called "blogging". It's very similar to the usual editing of your website, but still it's different because you get more features as a natural add-on: Categories, tagging, comments, publication dates, archives (yearly, monthly, daily), related entries, RSS feeds, etc.

On your website you simply go to the Blog area, and click on the "Add" link, which is available when you're logged in. In the blog entry editor you have all functionality you already know from the content management section. Modifying existing blog entries works identical to the usual content changes on your website with the exception of publishing, which is not available with an intermediate draft step: All your changes are immediately visible online.

Note that in django Organice we also use the blog functionality for other use cases that are very similar to the usual blogging, such as event agendas, job postings, press releases, download lists, etc.

3.3 Newsletter

Sending out news or updates to a circle of friends or customers is managed by the Newsletter component, which is available from the Django administration interface when you're logged in to your website. Simply click on "Admin > Site Administration" on the django CMS toolbar, and go to the Newsletter section in Django administration.

This component has a couple of powerful features. It's best explained having you watch the interesting French [overview video](#) from the Emencia website.

Contributing to django Organice

Official repositories: (kept in sync)

1. Bitbucket: <https://bitbucket.org/bittner/django-organice>
2. GitHub: <https://github.com/bittner/django-organice>

Fork any of the repositories, make your code changes or additions, and place a pull request.

4.1 Help Wanted

- Writing tests
- Translations (blog posts, documentation, user interface)

4.2 Bumping Versions, Package, Release

As for the version numbers of django Organice we use [Semantic Versioning](#).

Alternatives to django Organice

With django Organice we strive to make people happy who are not tech or process experts. People who want to collaborate smoothly without having taken part of extensive training programmes in order to understand the tool. Simplicity is key, everything should be intuitive, getting things done easily. Occasional users should be happy to the same extent than daily-habit expert users.

But of course, django Organice doesn't fit it all. If you feel it can't satisfy your specific needs it may be a good idea to look at the following alternatives. If you happen to know additional offers [drop us a note](#) so we can make the list more complete.

5.1 Software Alternatives

In alphabetical order.

- [Bitrix Intranet](#) (PHP, commercial, ships with source code)
- [Coyo](#) (Java/Play, commercial, ships with source code)
- [Microsoft SharePoint + Yammer](#) ¹ (.NET et al., commercial)

5.2 Cloud Competitors

In alphabetical order.

- [Bitrix24.com / Bitrix24.de](#) ¹
- [Igloo](#)

¹ <http://intranetsdoneright.blogspot.com/2013/11/what-is-social-intranet.html>

CHANGELOG

6.1 0.1

- Initial release
- Based on Django 1.5.5, django CMS 2.4.3, Zinnia blog 0.13, Emencia newsletter 0.3.dev
- A more natural i18n mechanism than vanilla Django, no language prefix for default language
- Setup script with project generation, deployment settings, custom templates, Bootstrap 3

Packages <https://github.com/bittner/django-organice/blob/v0.1/docs/requirements.txt>

Indices and tables

- *genindex*
- *modindex*
- *search*