wagtailosm Documentation

Release 0.1.0

Benjamin Bach

February 28, 2016

Contents

1 Features **3**

2 How it looks **5**

3 Improvements? **7**

3.1 Contents: 7

3.2 Feedback 10

Open Street Map integration for Wagtail

Features

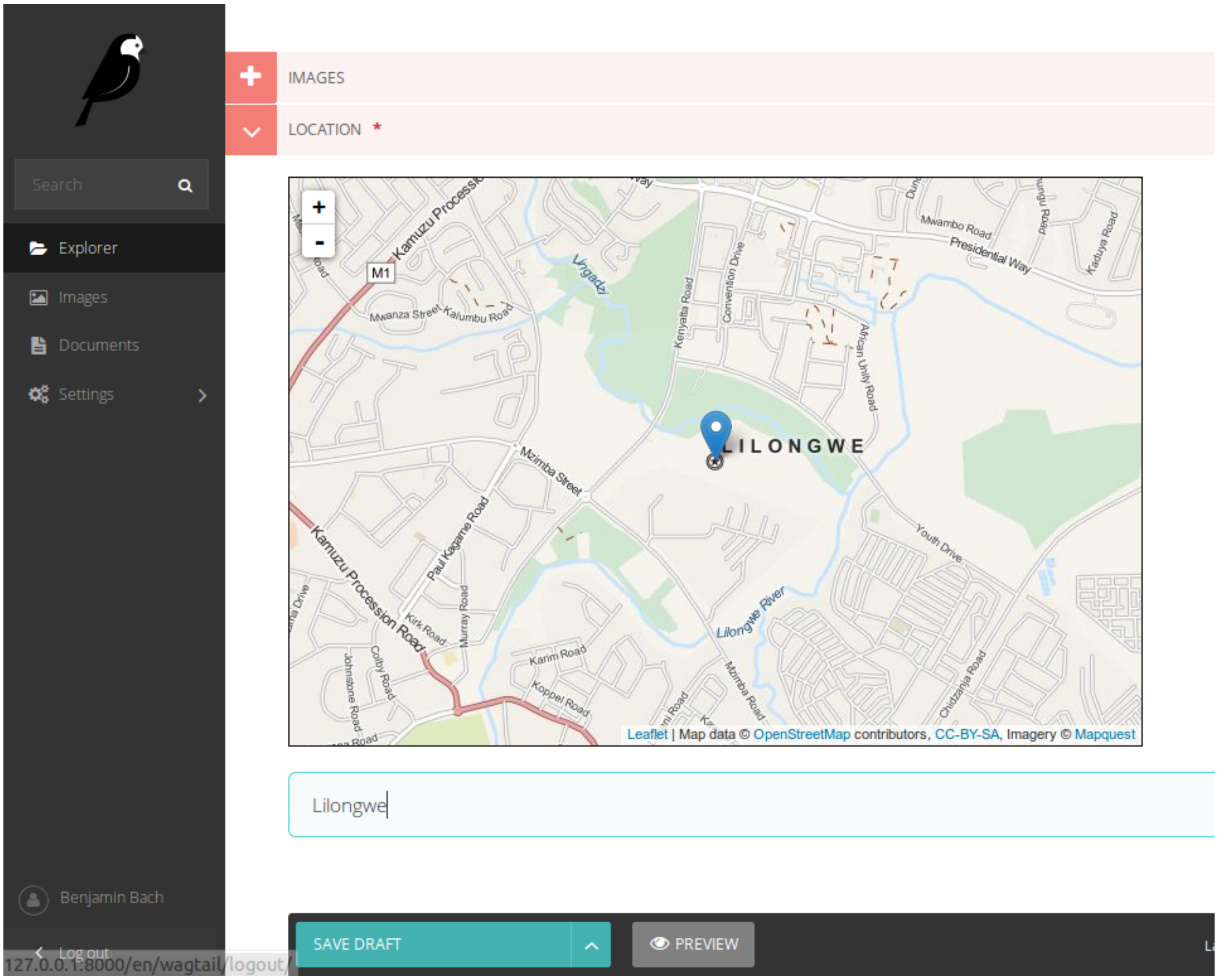
Simple integration of OpenStreetMaps in wagtail using the `osm_field` implementation.

Use it as with `osm_field`: If you want maps but don't need GIS.

Usage: See docs.

CHAPTER 2

How it looks



Lilongwe

Improvements?

This project took about 2 hours. Nothing is sacred, please contribute.

3.1 Contents:

3.1.1 Installation

At the command line either via easy_install or pip:

```
$ easy_install wagtailosm
$ pip install wagtailosm
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv wagtailosm
$ pip install wagtailosm
```

3.1.2 Usage

Add `wagtailosm` and `osm_field` to your `INSTALLED_APPS`:

```
INSTALLED_APPS = (
    # ...
    'osm_field',
    'wagtailosm',
    # ...
)
```

The order does not matter, but we need both of these applications to be registered because they contain static files.

To have a field with a map displayed in Wagtail, do this:

```
class Country(Page):

    # If you are adding this field to an existing model, you need to define some
    # default location for populating existing rows in the migration.
    DEFAULT_LAT, DEFAULT_LON = (52.4864, 13.4385)

    location = OSMField(default=(DEFAULT_LAT, DEFAULT_LON), lat_field='latitude', lon_field='longitu
    latitude = LatitudeField(default=DEFAULT_LAT)
    longitude = LongitudeField(default=DEFAULT_LON)
    # This is handled by the map, but don't forget you will probably need
    # to be able to customize this...
    zoom_level = models.PositiveSmallIntegerField(default=15)

    # Add to Wagtail admin's panels
    content_panels = Page.content_panels + [
        FieldPanel('location', widget=osm_field.fields.OSMWidget('latitude', 'longitude'), classname=
        FieldPanel('latitude', classname="wagtailosm-hidden"),
        FieldPanel('longitude', classname="wagtailosm-hidden"),
    ]
```

3.1.3 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

Types of Contributions

Report Bugs

Report bugs at https://github.com/benjaoming/wagtailosm/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

Write Documentation

wagtailosm could always use more documentation, whether as part of the official wagtailosm docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at https://github.com/benjaoming/wagtailosm/issues.

If you are proposing a feature:

- Explain in detail how it would work.

- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

Get Started!

Ready to contribute? Here's how to set up *wagtailosm* for local development.

1. Fork the *wagtailosm* repo on GitHub.
2. Clone your fork locally:

```
    $ git clone git@github.com:your_name_here/wagtailosm.git
```

3. Create a branch for local development:

```
    $ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

4. When you're done making changes, check that your changes pass style and unit tests, including testing other Python versions with tox:

```
    $ tox
```

To get tox, just pip install it.

5. Commit your changes and push your branch to GitHub:

```
    $ git add .
    $ git commit -m "Your detailed description of your changes."
    $ git push origin name-of-your-bugfix-or-feature
```

6. Submit a pull request through the GitHub website.

Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, and 3.3, and for PyPy. Check https://travis-ci.org/benjaoming/wagtailosm under pull requests for active pull requests or run the `tox` command and make sure that the tests pass for all supported Python versions.

Tips

To run a subset of tests:

```
$ py.test test/test_wagtailosm.py
```

3.1.4 Credits

Development Lead

- Benjamin Bach <benjamin@overtag.dk>

Contributors

None yet. Why not be the first?

3.1.5 History

0.1.0 (2015-12-26)

- First release on PyPI.

3.2 Feedback

If you encounter any errors or problems with **wagtailosm**, please let me know! Open an Issue at the GitHub http://github.com/benjaoming/wagtailosm main repository.