# django-leads Documentation

***Release 0.1.3***

**Diego Navarro Mellén**

December 05, 2013

# Contents

Contents:

CHAPTER 1

# django-leads

An easy, functional and customizable lead page for your next big thing, ready to use in your django project.

## 1.1 Documentation

The full documentation is at http://django-leads.rtfd.org.

## 1.2 Quickstart

### 1.2.1 Install django-leads

```
$ pip install django-leads  # Also auto installs all needed dependencies! :)
```

Or get the bleeding-edge version:

```
$ pip install https://github.com/dnmellen/django-leads/archive/master.zip
```

### 1.2.2 Settings

```
INSTALLED_APPS = (
    ...
    'floppyforms',
    'crispy_forms',
    'leads',
)
...
CRISPY_TEMPLATE_PACK = 'bootstrap3'
```

### 1.2.3 urls.py

```
...
import leads.urls

urlpatterns = patterns('',
    ...
    url(r'^admin/', include(admin.site.urls)),
    url(r'^', include(leads.urls, namespace='leads')),
)
```

### 1.2.4 Final touch

```
$ python manage.py syncdb
```

## 1.3 Features

- Works on Python 2.6, 2.7, 3.3
- Basic lead page to act as a placeholder for your django project
- Registers user's name and email into database
- Customizable and extendable models and forms
- Admin interface
- Export registered users to CSV
- Send newsletters to your users

CHAPTER 2

# Installation

At the command line:

```
$ easy_install django-leads
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv django-leads
$ pip install django-leads
```

CHAPTER 3

# Usage

## 3.1 Start a django project from scratch

### 3.1.1 Create virtualenv

```
$ mkdir test_lead
$ cd test_lead
$ mkvirtualenv leads
```

### 3.1.2 Install from pip

```
$ pip install django-leads
```

### 3.1.3 Start a brand new django project

```
$ django-admin.py startproject mysite
```

Edit *settings.py*

```
INSTALLED_APPS = (
    ...
    'floppyforms',
    'crispy_forms',
    'leads',
)
...
CRISPY_TEMPLATE_PACK = 'bootstrap3'
```

Edit *urls.py*

```
...
import leads.urls

urlpatterns = patterns('',
    ...
    url(r'^admin/', include(admin.site.urls)),
```

```
    url(r'^', include(leads.urls, namespace='leads')),
)
```

Build database

```
$ python manage.py syncdb
```

Launch server

```
$ python manage.py runserver
```

## 3.2 Overriding templates

To have a reference you can copy the current templates:

```
$ cd <your_django_project_path>
$ mkdir templates
$ cd templates
$ cp -r ~/.virtualenvs/<your_virtualenv>/lib/python2.7/site-packages/leads/templates/leads .
```

Make sure you can load templates from `<your_django_project_path>/templates` (check your `settings.py`):

```
PROJECT_PATH = os.path.dirname(os.path.realpath(__file__))
TEMPLATE_DIRS = (
    os.path.join(PROJECT_PATH, '..', 'templates'),
)
```

Now you can override any template contained in `../templates/leads`

## 3.3 Advanced settings

*django-leads* allows the user to set some settings to change the default behaviour

| Setting name | Explanation |
|---|---|
| LEADS_REGISTER_MODEL | Define a custom Model to save data |
| LEADS_REGISTER_MODEL_ADMIN | Define a custom ModelAdmin class |
| LEADS_REGISTER_FORM_FIELDS | Specify the fields that will be shown in the form |
| LEADS_REGISTER_FORM_CLASS | Define a custom FormModel to represent the form |

Example:

```
# settings.py
LEADS_REGISTER_FORM_CLASS = 'myapp.forms.CustomRegisterForm'
LEADS_REGISTER_FORM_FIELDS = ('email',)
```

CHAPTER 4

# 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:

## 4.1 Types of Contributions

### 4.1.1 Report Bugs

Report bugs at https://github.com/dnmellen/django-leads/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.

### 4.1.2 Fix Bugs

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

### 4.1.3 Implement Features

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

### 4.1.4 Write Documentation

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

### 4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/dnmellen/django-leads/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 :)

## 4.2 Get Started!

Ready to contribute? Here's how to set up *django-leads* for local development.

1. Fork the *django-leads* repo on GitHub.
2. Clone your fork locally:

   ```
   $ git clone git@github.com:your_name_here/django-leads.git
   ```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

   ```
   $ mkvirtualenv django-leads
   $ cd django-leads/
   $ python setup.py develop
   ```

4. Create a branch for local development:

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

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 leads tests
    $ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. 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
   ```

7. Submit a pull request through the GitHub website.

## 4.3 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/dnmellen/django-leads/pull_requests and make sure that the tests pass for all supported Python versions.

## 4.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_leads
```

CHAPTER 5

# Credits

## 5.1 Development Lead

- Diego Navarro Mellén <dnmellen@gmail.com>

## 5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

# History

## 6.1 0.1.0 (2013-11-08)

- First release on PyPI.

## 6.2 0.1.1 (2013-11-10)

- Makes this app more extendable by using custom models and forms

## 6.3 0.1.2 (2013-12-04)

- Exports registered users to CSV from admin panel
- Create newsletters from admin panel
- Fixed installation from pip

## 6.4 0.1.3 (2013-12-04)

- Bugfixed circular imports when setting custom Forms