TailorDev Contact Documentation

Release 0.5.0

Julien Maupetit

January 12, 2015

Contents

Contents:

CHAPTER 1

Django TailorDev Contact

A customizable contact form for your django projects.

1.1 Dependencies

For now, Django>=1.5 is the only dependency for this project to run on production, with python>=2.6. Currently, this application is not compatible with python 3.3. We are working on it.

1.2 Installation

To install TailorDev Contact, use pip:

```
$ pip install django-tailordev-contact
```

If you intend to test or improve this application, first clone this repository and install the local dependencies:

```
$ pip install -r requirements/local.txt
```

1.3 Configuration

Add `td_contact` and its dependencies to your `INSTALLED_APPS`:

```
INSTALLED_APPS = (
...
    'td_contact',
...
)
```

Add `td_contact` urls to your project url patterns:

```
urlpatterns = patterns('',
    ...
    url(r'^contact/', include('td_contact.urls')),
    ...
)
```

Set your `td_contact` rules in your `settings.py`, by adding something like:

```
# Contact form
TD_CONTACT_FORM_RULES = {
    'default': {
        'prefix': "[Foo:contact]",
        'subject': "General informations",
        'to': ('contact@foo.com', 'ceo@foo.com'),
    },
    'partner': {
        'prefix': "[Foo:partner]",
        'subject': "Partnership opportunity",
        'to': ('partner@foo.com', ),
    },
    'jobs': {
        'prefix': "[Foo:jobs]",
        'subject': "Job opportunity",
        'to': ('jobs@foo.com', ),
    },
}
```

`TD_CONTACT_FORM_RULES` is a simple dictionary where each key defines a new rule. Each rule is also a dictionary defining the email prefix & subject and the recipient list.

> **Important note** : when a contact form has been successfully filled, the user is redirected to the website home page. Thereby, we use the django messages framework to inform our user of its request status. Remember to enable messages and add something like the following to your base template DOM:

```
{% if messages %}
<ul class="messages">
    {% for message in messages %}
    <li data-alert{% if message.tags %} class="message {{ message.tags }}"{% endif %}>
        {{ message }}
        <a href="#" class="close">&times;</a>
    </li>
    {% endfor %}
</ul>
{% endif %}
```

This example works with the zurb foundation framework. Feel free to adapt this for your favorite framework.

1.4 Urls

TailorDev Contact form defines 3 urls you may use in your templates:

1.4.1 `contact_form_rule`

This url has been designed to initialize your form with your own rule, *e.g.*:

```
{% url 'contact_form_rule' 'jobs' %}
```

1.4.2 `user_contact_form_by_slug` and `user_contact_form_by_pk`

Depending on your application, people may want to contact a registered user directly and you want an elegant url to point to. To do so, use the `user_contact_form_by_slug` in your templates, *e.g.*:

```
{% url 'user_contact_form_by_slug' myuser.slug %}
```

Alternatively, use the `user_contact_form_by_id` in your templates, like:

```
{% url 'user_contact_form_by_pk' myuser.pk %}
```

1.4.3 `contact_form`

This base url points to your contact form. Nothing more to add.

1.5 Templates

1.5.1 Using the default templates

If you want to use our default templates, feel free to do so. But please note that:

- You should create a base template to inherit from, visible as `_layouts/base.html`
- Your form will appear in a `content` block
- Two partial templates must be customized `contact/partials/contact_recipient.html` and `contact/partials/aside.html`

1.5.2 Using your own template(s)

The template-to-override used to render the form is visible as `contact/form.html`. The core part of the template may looks like:

```
<div class="form_wrapper">

    <h1>{% trans "Contact" %}</h1>

    {% if recipient %}
        {% include "contact/partials/contact_recipient.html" %}
    {% endif %}

    <form action="" method="post" class="custom">
        {% csrf_token %}
        {% for field in form %}
        {% if field.is_hidden %}
        {{ field }}
        {% else %}
        <div class="field_wrapper">
            <div class="field{% if field.field.required %} required{% endif %}">
                <label {% if field.errors %}class="error"{% endif %}>{{ field.label }}</label>
                {{ field }}
                {% if field.errors %}
                <small class="error">{{ field.errors }}</small>
                {% endif %}
            </div>
        </div>
        {% endif %}
        {% endfor %}
```

```
        <button type="submit" />{% trans "Send message" %}</button>
    </form>
</div>
```

1.6 Running the Tests

You can run the tests with via:

```
python setup.py test
```

or:

```
python runtests.py
```

1.6.1 Code coverage

To estimate the project coverage:

```
coverage run --source='td_contact' runtests.py
coverage report -m
```