
phileo Documentation

Release 0.1

Eldarion

October 27, 2011

CONTENTS

A user likes app.

DEVELOPMENT

The source repository can be found at <https://github.com/eldarion/phileo>

1.1 Contents

1.1.1 Installation

- To install phileo:

```
pip install phileo
```

- Add 'phileo' to your INSTALLED_APPS setting:

```
INSTALLED_APPS = (  
    # other apps  
    "phileo",  
)
```

- Lastly you will want to add *phileo.urls* to your urls definition:

```
...  
url(r"^likes/", include("phileo.urls")),  
...
```

1.1.2 Usage

Phileo consists of template tags that you place within your project to get different “liking” functionality.

like_form

This template tag is an inclusion tag that will emit a form that renders a button that enables toggling the “like” or “unlike”. The button will have the text that you pass in as well as a class by the same name, but all lower case. This should enable you to control the look of the button through css.

The format for this template tag is:

```
{% like_form user object "Like,Unlike" %}
```

The string of the toggled states that you pass in should follow some rules:

- Only contain two verb labels

- First label is the initial action you want to offer users
- The casing of the text here is exactly how it will appear in the button

It renders “phileo/_form.html” and can be overridden as desired.

likes_ajax

This is a simple inclusion template tag that will render the a bit of javascript for doing the form post and changing the form state via ajax. If you do not include this, then the button on the form from like_form will still work, but will just be a form post back returning you to the same page:

```
{% likes_ajax %}
```

It renders “phileo/_ajax.js” and can be overridden as desired.

liked

The “liked” template tag will decorate a iterable of objects given a particular user, with a “liked” boolean indicating whether or not the user likes each object in the iterable:

```
{% liked objects by request.user as varname %}
{% for obj in varname %}
    <div>{% if obj.liked %}* {% endif %}{{ obj.title }}</div>
{% endfor %}
```

likes

The “likes” tag will fetch into a context variable a list of objects that the given user likes:

```
{% likes request.user "app.Model" as objs %}
{% for obj in objs %}
    <div>{{ obj }}</div>
{% endfor %}
```

1.1.3 Signals

Both of these signals are sent from the Like model in the view that processes the toggling of likes and unlikes.

phileo.signals.object_liked

This signal is sent immediately after the object is liked and provides the single kwarg of “like” which is the instance of the Like object that was created.

phileo.signals.object_unliked

This signal is sent immediately after the object is unliked and provides the single kwarg of “object” which is the objects that was just unliked.

1.1.4 ChangeLog

0.1

- initial release