pinax$_t$heme$_f$oundationDocumentation

Release 0.2.3

ChrisDev

August 30, 2016

A Pinax theme based on Zurb Foundation. Foundation is a popular CSS framework that is light weight, but includes all the basics such as; a responsive grid, forms, dialog, navigation tabs, buttons, typography and so on. You can read more about the ideas behind Foundation and how to use it for rapid prototyping in this article.

Contributors

- Christopher Clarke
- Lendl R Smith
- Kewsi Aguillera
- Justin Hamade

1.1 What's New

This version supports Foundation 5

- We support the latest version of Foundation (5.2.1)

- Updated the documentation

- The project now supports the `--template` argument for `django-admin.py startproject` command

- Removed more `Pinax 0.9` dependencies however, the project should still be compatible with the other apps in the `Pinax` eco-system such as django-user-account

Getting Started

Start by creating a new virtualenv for your project and install Django 1.6.2

```
mkvirtualenv mysite
pip install Django==1.6.2
```

To use create a project based on the zero project run

```
django-admin.py startproject mysite --template=https://github.com/chrisdev/foundation-project-zero/z:
```

This will create a new Django project in the mysite directory with:

- An apps folder for your internal apps
- initial data (for handling `sites.Site` model)
- Requirements files for use with pip
- A home for your site's static files
- A set of templates designed to work with the foundation theme

Finally install the requirements, sync your database and start the test server

```
cd mysite
pip install -r requirements.txt
python manage.py syncdb && python manage.py runserver
```

The account starter project provides all the features of the zero project in addition to incorporating features to support django-user-account.

django-user-account is an extremely useful `Pinax` app that works in conjunction with django.contrib.auth to take your Django project from having simple log in, log out and password reset to a full blown account management system. To create a project based on the account starter project

```
django-admin.py startproject --template=https://github.com/chrisdev/foundation-project-account/zipbal
```

Templates

Your own templates should normally inherit from `site_base.html`. You should provide your own "footer" template `_footer.html`.

Also change the Site name by editing `fixture/initial_data.json`. You can also use the Admin app for this purpose.

The **url** name "home" should be defined as the homepage.

Upgrading From Previous Versions

To upgrade your site start by upgrading to the latest version on pinax-theme-foundation

```
pip install -- upgrade pinax-theme-foundation
```

The big change between Foundation 2 to 3 is the grid. In Foundation 3 you no longer have to use *.container* to define the grid. In Foundation 2 the grid was built around *.container > .row > .columns*. In Foundation 3 you now just need to use *.row > .columns*. In Foundation 3 padding and borders no longer increase the width of an element, they only go inward so for example in Foundation 3 .three.columns always has a width of 25% with a 15px padding on the left and right. You can find the Foundation 3 migration guide here

The Details

6.1 Top Navbar

Foundation 3 now includes a top nav bar includes the main nav options as well as hover dropdowns that support either a list of anchors or arbitrary content (even Grid content).

To use the *Top Navbar* in your your project, first ensure that your project's *site_base.html* inherits from *theme_base.html*

```
{% extends "theme_base.html" %}
```

theme_base.html defines the template block *topbar* with the following structure.

```
{% block topbar_base %}
    <div class="attached">
            <div class="row">
                    <li class="name mobile-four">
                            <a href="/">{{SITE_NAME}}</a><a href="#"></a>
                    </li>
                    <ul class="nav-bar menu mobile-four">
                            {% block topbar %}
                                    {% block nav %}{% endblock %}
                                    {% block account_bar %}
                                            {% include "_account_bar.html" %}
                                    {% endblock %}
                            {% endblock %}
                    </ul>
            </div>
    </div>
{% endblock %}
```

This will provide your project with a top *nav bar* which is attached to the top of the page that displays the *SITE_NAME* and an account management drop down menu. The account management drop down is simply an unordered lists as shown in the following snippet abstracted from the *_account_bar.html* template file

```
{% if user.is_authenticated %}
    <li class="has-flyout">
        <a href="#">{{ user }}</a>
        <a href="#" class="flyout-toggle"><span> </span></a>
        <ul class="flyout">
            <li><a href="{% url acct_email %}"><span class="glyph social">x</span>Account</a></i></li
            {% if user.is_staff %}
                    <li><a href="{% url admin:index %}"><span class="glyph general">a</span>Admin</a>
```

```
            {% endif %}
            <li><a href="{% url acct_logout %}"><span class="glyph general">]</span>Log out</a><
        </ul>
    </li>
{% endif %}
```

So to create a drop down menu item, simply assign the *li* element the *"has-flyout"* class and include an unordered list element with a "flyout" class selector `<ul class="flyout">` that contains the drop down menu items . The *nav* block can be utilized for the other menu items for the site. These can be a combination of drop down and non drop down elements. This is shown in the following snippet abstracted from the *Top Navbar* for this site

```
<li><a href="/">Home</a></li>
    {% if user.is_authenticated %}
<li id="tab_profile"><a href="{% url profile_detail user.username %}">{% trans "Profile" %}</a></li>
<li id="tab_notices"><a href="{% url notification_notices %}">{% trans "Notices" %}{% if notice_unsee
    {% endif %}
```

It should be noted that menu items can be embellished with appropriate symbol icons by including a span element with the *"glyph general"* class selector. So adding :

```
<span class="glyph general">a</span>
```

will display the "gear" icon. These are actually Zurb symbol icon font that will scale and display nicely on various devices. These can be used any where in your project.

6.2 Forms

This theme ships with a basic template tag for rendering forms that match the markup expected by foundation.

To style forms, add the following to the top of your template

```
{% load foundation_tags %}
```

and include your form using the following markup:

```
<form method="POST" action="">
    {% csrf_token %}
    <fieldset class="form-controls">
        {{ form|as_foundation }}
    </fieldset>
    <fieldset class="form-actions">
        <button type="submit" class="button medium radius nice"">Submit</button>
    </fieldset>
</form>
```

Find more about foundation forms here .

6.3 Modal Dialogs

Reveal is foundation's modal dialog The easiest way to use reveal dialogs with your *pinax_* forms is to first create a separate template for the form as shown in this simplified example taken from the idios app.

```
    {% extends "idios/base.html" %}
{% load foundations_tags %}
    {% load i18n %}
```

```
    {% block head_title %}{% trans "Edit Profile" %}{% endblock %}


    {% block body %}
    <form class="nice" method="POST" action="{% url profile_edit %}">
            {% csrf_token %}
            <fieldset>
                    {{ profile_form|as_foundation }}
            </fieldset>
            <div class="actions">
                    <button type="submit" class="button medium radius nice">Update</button>
            </div>
    </form>
    {% endblock %}
```

In the template (*idios/profile.html*) where you to show the modal form, add the *"reveal"* class selector and give it an id (*edit-profile-box*) to the object which you want to fire the modal when clicked:

```
<a id="edit-profile-box" href="{% url profile_edit %}" class="button reveal">Edit profile</a>
```

The *href* `{% url profile_edit %}` url will display the form. Finally add the following javascript to the template.

```
{% block extra_body %}
<script>
   $('#edit-profile-box').click(function(event) {
               event.preventDefault();
               var $div = $('<div>').addClass('reveal-modal').appendTo('body');
               $this = $(this);
               $.get($this.attr('href'), function(data) {
                 return $div.empty().html(data).append('<a class="close-reveal-modal">&#215;</a>').r
               });
});
</script>
{% endblock %}
```

It should be noted that

6.4 Mobile Visibility

Zurb Foundation includes a series of visibility classes that allow you to hide or display content on various devices using 'show/hide-for-x' where "x" can be a xlarge, large, medium and small. For example if we include the following snippet on the line corresponding to the device that you are currently using will be displayed

```
<strong class="show-for-xlarge">You are on a very large screen.</strong>
<strong class="show-for-large">You are on a large screen.</strong>
<strong class="show-for-large-up">You are on a large or very large screen.</strong>
<strong class="show-for-medium">You are on a medium screen.</strong>
<strong class="show-for-medium-down">You are on a medium or small screen.</strong>
<strong class="show-for-small">You are on a small screen, like a smartphone.</strong>
```

Conversely including

```
<strong class="hide-for-xlarge">You are not on a very large screen.</strong>
<strong class="hide-for-large">You are not on a large screen.</strong>
<strong class="hide-for-large-up">You are not on a large or very large screen.</strong>
```

```
<strong class="hide-for-medium-down">You are not on a medium or small screen.</strong>
<strong class="hide-for-medium">You are not on a medium screen.</strong>
<strong class="hide-for-small">You are not on a small screen.</strong>
```

Will display the two lines corresponding to the devices that you are not using.

6.5 CHANGELOG

6.5.1 0.2.3 (2013/02/26)

Changes are mainly from our new contributor Justin Hamade

- Updated all the foundation css classes and included support for the new icon fonts To create an icon apply the appropriate classes that match the icon you want to use.

```
<i class="foundicon-[icon]"></i>
```

- Added demos from the ZURB playground, these make a nice visual test to make sure things were working:

    - Responsive Tables

    - Off canvas layouts

    - Icon Fonts 2

- Added all of the templates from http://foundation.zurb.com/templates.php

- Removed minimized ccs/js in favour of using an inline compressor like django-pipeline. This is has currently been implemented in Account Template

- Fixes for the orbit slider

6.5.2 0.2a2 (2012/11/02)

- Provide support for the –*template* for `django-admin.py startproject`

- Removed `Pinax` dependency. Note it will still work with other pinax style apps such as django-user-account

6.5.3 0.1.7 (2012/10/18)

- This release supports Foundation 3.1 which includes features such as right-to-left language support, new UI Styles for Progress Bars and Image Thumbs, updated jQuery and so on, read more Foundation 3.1 here

- Replaced any of the remnants of our home grown Top Navigation menu with the Foundation 3.1 responsive Top Navigation bar

- Included Icon Fonts, Responsive Tables and SVG Social Icons which are not part of core release

- Lay the groundwork for supporting the –*template* flag on the *django-admin.py start project* in the next release

- Other Bug fixes.

6.5.4 0.1.6 (2012/08/02)

- We now support Foundation 3 so you can use features like new the new Top Foundation Nav bar

- Support for responsive tables

- Simplified the demo site so that its just the pinax basic site

- Removed support for responsive design patterns as these were not compatible with Foundation 3 release

- Updated documentation to reflect Foundation 3

- Numerous bug fixes.

6.5.5 0.1.5 (2012/06/08)

- Improved top Navbar based on which is based on foundation's top-bar branch .

- Inclusion of zurb symbol icon fonts.

- Inclusion of CSS to support the Responsive Design Patterns originally discussed by Joshnua Johson and implemented by Matt Reimer

- Updated documentation and set up a demo site based on a Pinax basic project

- Some reorganization of the *theme_base.html* including:

    - Moved (where possible) the javascript to the bottom of the file

    - Use the static template tag

    - Using the static assets for released version of zurb-foundation 2.2.1

- Fixed numerous bugs

6.5.6 0.1.4 (2012/03/004)

- Added the Fixed top Menubar

- Show active links on the menubar (Issue #14)

- Fixed the repositioning of form when timezone is changed on timezone page

- Removed unnecessary cancel button on the modal Profile form

- Updated the Foundation theme to Zurb Foundation 2.2.0

6.5.7 0.1.3b (2012/2/13)

- Added Mobile Login/Sign in functionality

6.5.8 0.1.3

- Initial release of pinax-theme-foundation