# Contents

1. About Bambu Bootstrap 3
2. About Bambu Tools 2.0 5
3. Installation 7
4. Basic usage 9
5. Questions or suggestions? 11
6. Indices and tables 19

Python Module Index 21
Use Twitter’s Bootstrap CSS framework to build your app. All the views Bambu uses all extend a base template which you create, that can be based on a skeleton Bootstrap template. Shortcut tags let you easily add breadcrumb trails and icons to your apps.
Bambu Tools is a set of reusable Django apps and utility packages that help prototyping and building web apps easier. To a degree, this starts with the front-end scaffolding. Bambu Bootstrap provides a base template along with a set of useful tags and filters that make building web apps using this framework easier.
About Bambu Tools 2.0

This is part of a toolset called Bambu Tools. It’s being moved from a namespace of `bambu` to its own ‘root-level’ package, along with all the other tools in the set. If you’re upgrading from a version prior to 2.0, please make sure to update your code to use `bambu_bootstrap` rather than `bambu.bootstrap`. 
Install the package via Pip:

```
pip install bambu-bootstrap
```

Add it to your `INSTALLED_APPS` list and add Bootstrap and Font-Awesome to your `BOWER_INSTALLED_APPS` list (see the django-bower documentation) for details on managing static files through Bower.

```
INSTALLED_APPS = {
    'djangobower',
    'bambu_bootstrap'
}
```

```
BOWER_INSTALLED_APPS = {
    'bootstrap',
    'fontawesome'
}
```

Remember to run `python manage.py bower install` and `python manage.py collectstatic`. 
CHAPTER 4

Basic usage

Within your project, start with a template called base.html. This should extend the Bootstrap base template, at bootstrap/base.html.

Use the extra_head block to specify stylesheets or script tags that must, by necessity live in the head of your document.

Use the content block for the main content of your page.

Use the javascript block to specify JavaScript that can run at the very bottom of the page.
Questions or suggestions?

Find me on Twitter (@iamsteadman) or visit my blog.

Templates

This is where the main heart of Bambu Bootstrap can be found.

Error pages

These two templates handle Django’s 404 and 500 error pages in a friendly manner. They’re meant to be copied and messed with, so grab them from the source code and amend them, or just roll your own.

404.html

A Bootstrap-themed 404 Not Found page.

500.html

A Bootstrap-themed 500 Internal Server Error page.

Forms

See the Forms section for more information.

field.inc.html

An individual field row, with error checking and handling of various different styling scenarios.
A rendered form, using the context variable form. By default the form is rendered in Bootstrap’s ‘vertical’ style, but you can use the horizontal style by adding style='horizontal' in your include statement.

Blocks:

• non_field_errors: A list of general form errors, not relating to specific fields, rendered using Bootstrap’s alert classes.
• fields: The list of fields (except for the DELETE field, ie: when used as part of a fieldset). By default, this block is rendered by looping through the visible fields in the form and including the field.inc.html template.
• fieldset: Houses the DELETE field for a form within a formset
• hidden_fields: The remaining, invisible fields of the form

**Navbar**

This is arguably the most involved part of the Bambu Bootstrap template structure. You might want to strip much of this out once your app is up and running, to save a little overhead.

**bootstrap(navbar-anon.inc.html**

A horizontal login form. Included in bootstrap/navbar.inc.html for anonymous users.

**bootstrap(navbar-auth.inc.html**

Renders a dropdown menu with a logout link. Included in bootstrap/navbar.inc.html for authenticated users.

**bootstrap(navbar-brand.inc.html**

Renders the site name with a link to the homepage, with Bootstrap’s navbar-brand class.

**bootstrap(navbar.inc.html**

The main top navigation bar for a site. It includes the templates above (subject to their conditions) and menu.inc.html.

**menu.inc.html**

The header menu. Copy this template from the source code or start with the following to add a menu:

```html
<ul class="nav navbar-nav">
    <li><a href="/">Home</a></li>
</ul>
```
Elements

Here are a few miscellaneous elements

**bootstrap/base.inc.html**

The engine room. This template includes the HTML5 doctype and tags. Your own base template should extend this one, using blocks to fill in the content.

Blocks:

- **title**: The title tag for a page (see the `bootstrap_title` template tag for more information)
- **meta_tags**: The meta tags for a page
- **extra_head**: Where your base template should place its `<head>` tag content
- **after_header**: The first thing displayed after the navigation bar
- **messages**: The block that includes the `bootstrap/messages.inc.html` template
- **breadcrumb_trail**: The block that includes the `bootstrap/breadcrumb.inc.html` template
- **content**: Where the main page content lives
  - **page_header**: The content for the Bootstrap `.page-header` element
  - **form_content**: The primary (left-hand) column of content
  - **sidebar**: The (right-hand) sidebar
  - **form_footer**: The element to display after the form-content and sidebar
- **pre_footer**: Appears straight after the main content block
- **footer**: Appears inside a `<footer>` tag, and includes the `bootstrap/footer.inc.html` template
- **javascript**: The area to place all JavaScript content that is safe to include at the bottom of a page

**bootstrap/breadcrumb.inc.html**

A breadcrumb bar. See the `context variable` or `template tag` for more information.

**bootstrap/footer.inc.html**

The page footer. By default this is a simple copyright statement using the site name and current year.

**bootstrap/messages.inc.html**

Bootstrap-themed alert boxes for messages generated by `django.contrib.messages`.

**bootstrap/scripts.inc.html**

Includes the jQuery and Bootstrap script tags. Use the `BOOTSTRAP_JS_URL` setting to provide your own Bootstrap script.
bootstrap/styles.inc.html

Includes the Bootstrap and Font Awesome CSS link tags. Use the `BOOTSTRAP_CSS_URL` setting to provide your own Bootstrap theme.

bootstrap/typekit.inc.html

Includes the TypeKit script tag. See the `TYPEKIT_KEY` setting.

Search

Bambu Bootstrap also works well with Haystack, and provides a basic search results template.

search/search.html

A basic search results template including a form.

Context variables

Bambu Bootstrap recognises a number of context variables, passed in via a TemplateResponse object. Their names are deliberately kept generic so as not to enforce a drastic change should you want to move away from Bootstrap.

body_classes

A tuple of classes to add to the page’s `<body>` tag. Can be generated by the `body_classes` decorator.

breadcrumb_trail

A two-tuple containing the URL and title of each item in a breadcrumb trail. The first item in the list should be the homepage, the last item (whose URL should be an empty string) should be the current page.

Example:

```python
context['breadcrumb_trail'] = (  
    ('../', u'Home'),  
    ('..', u'The parent page'),  
    ('', u'This page')  
)
```

This value is rendered by the `breadcrumb_trail` template tag.

title_parts

A tuple that forms the text for a page’s `<title>` tag. The `bootstrap_title` template tag separates each part with a beam (`|`) character, and adds the name of the site to the end.
meta_keywords

Text for the value of a page's `<meta name="keywords">` tag.

meta_description

Text for the value of a page's `<meta name="description">` tag.

popup

Defines whether the current page is a popup window. Typically you'd set this context variable in a view by checking for something in the querystring. If set, it adds `popup` to the list of classes in the `<body>` tag.

Views

A base view with a number of mixins are provided here, as well as a simple template view that provides an easy `extra_context` property.

```python
class bambu_bootstrap.views.BootstrapView(**kwargs)
    A class-based view to be rendered via a Bootstrap template, providing ways to setup `<body>` tag classes, formula for the `<title>` tag, the breadcrumb trail and a key indicating the selected main navigation item item.

    get_base_context(request, **kwargs)
        Sets up the base context for the templated view

class bambu_bootstrap.views.DirectTemplateView(**kwargs)
    This is similar to Django’s old direct template generic view. It’s handiest when used for ‘static’ pages like homepages (ie: where dynamic data may come from context processors so a standard view isn’t needed). It supports context variables via the `extra_context` argument.

class bambu_bootstrap.views.FormMixin
    A view mixin that provides a form for saving data.

        form_class
            alias of Form

        get_form(request, **kwargs)
            Instantiates the form

        redirect_success(request)
            Redirects back to the currently-requested URL

        save_form(request, form)
            Saves the form data and returns the saved object

        validate_form(request, form)
            Checks that the form data is valid

class bambu_bootstrap.views.MessageMixin
    A view mixin that provides a simple way to implement django.contrib.messages. You can define messages for various labels (‘info’, ‘success’, ‘warning’, ‘error’) and send them via a simple function.

        message(request, key)
            Sends a message to a user
```
class bambu_bootstrap.views.ObjectMixin
   A view mixin that makes it easier to work with single object views, where the title and breadcrumb trail might be influenced by a common factor.

Forms

Date fields

Add bootstrap-datepicker to your BOWER_INSTALLED_APPS setting.

Link up the CSS via the extra_head block:

```html
<link href="{% static 'bootstrap-datepicker/css/datepicker3.css' %}" rel="stylesheet" media="screen" />
```

And add the script via the javascript block:

```html
<script src="{% static 'bootstrap-datepicker/js/bootstrap-datepicker.js' %}'></script>
```

Font Awesome

Bambu Bootstrap maintains a list of icon classes provided by the Font Awesome library.

Context processor

Bambu Bootstrap has a context processor that returns the current site, as defined in the Django sites framework.

Installation

Add the following to your list of processors:

```python
bambu_bootstrap.context_processors.basics
```

Usage

Use the SITE context variable to reference the current site’s name or domain.

Decorators

bambu_bootstrap.decorators.body_classes(func, *classes)
   Decorates a view by adding a list of class names to the <body> tag. The preferred way to use this is to pass in a view that returns a TemplateResponse object, so that the decorator can modify the context variable dictionary, adding a body_classes list, or extending it if it already exists.

   Parameters

   • func – A callable that returns an HttpResponse or TemplateResponse object
• **classes** – A list of classes to add to the `<body>` tag

Use this decorator in your URLconf like so:

```python
from bambu_bootstrap.decorators import body_classes
from testproject.myapp import views
from django.conf.urls import patterns, url

urlpatterns = patterns('',
    url(r'^$', body_classes(views.home, 'homepage', 'index'))
)
```

## Template tags

### Settings

**BOOTSTRAP_CSS_URL**: The URL (relative to the `MEDIA_URL` setting) of a custom Bootstrap CSS build

**BOOTSTRAP_JS_URL**: The URL (relative to the `MEDIA_URL` setting) of a custom Bootstrap JavaScript build

**TYPEKIT_KEY**: Your TypeKit API key

**BOOTSTRAP_NAVBAR_INVERSE**: Apply the `navbar-inverse` class to the main navigation bar (default is `False`)

**BOOTSTRAP_NAVBAR_FIXED_TOP**: Fix the navigation bar to the top. You should add padding to your `body` tag via CSS to compensate for the effect of adding fixed-positioning to the navbar (default is `False`).

**BOOTSTRAP_NAVBAR_FIXED_BOTTOM**: Fix the navigation bar to the bottom. You should add padding to your `body` tag via CSS to compensate for the effect of adding fixed-positioning to the navbar (default is `False`).
CHAPTER 6

Indices and tables

- genindex
- modindex
- search
bambu_bootstrap.decorators, 16
bambu_bootstrap-fontawesome, 16
bambu_bootstrap.templatetags.forms, 17
bambu_bootstrap.templatetags.icons, 17
bambu_bootstrap.views, 15
Index

B
bambu_bootstrap.decorators (module), 16
bambu_bootstrap.fontawesome (module), 16
bambu_bootstrap.templatetags.forms (module), 17
bambu_bootstrap.templatetags.icons (module), 17
bambu_bootstrap.views (module), 15
body_classes() (in module bambu_bootstrap.decorators), 16
BootstrapView (class in bambu_bootstrap.views), 15

D
DirectTemplateView (class in bambu_bootstrap.views), 15

F
form_class (bambu_bootstrap.views.FormMixin attribute), 15
FormMixin (class in bambu_bootstrap.views), 15

G
get_base_context() (bambu_bootstrap.views.BootstrapView method), 15
get_form() (bambu_bootstrap.views.FormMixin method), 15

M
message() (bambu_bootstrap.views.MessageMixin method), 15
MessageMixin (class in bambu_bootstrap.views), 15

O
ObjectMixin (class in bambu_bootstrap.views), 15

R
redirect_success() (bambu_bootstrap.views.FormMixin method), 15

S
save_form() (bambu_bootstrap.views.FormMixin method), 15

V
validate_form() (bambu_bootstrap.views.FormMixin method), 15