django-bootstrap-ui simplifies the integration of the popular Bootstrap UI framework. It is written in Python and built on the Django web framework.
CHAPTER 1

Basic features

• Basic HTML5 template
• Extensive and up-to-date documentation
• Mainstream Python and Django support
• Outstanding test coverage
• Continuously integrated codebase
CHAPTER 2

Bootstrap 4 features

- Full-featured Bootstrap 4 template (4.6.1)
- Integrated Google material design icon font
CHAPTER 3

Bootstrap 3 features

- Full-featured Bootstrap 3 template (3.3.7)
- Latest Font Awesome integration (4.7.0)
- Built-in Bootstrap and Bootswatch themes (3.3.7)
Technical requirements

Below is the list of currently supported combinations of Django and Python:

<table>
<thead>
<tr>
<th></th>
<th>Python</th>
<th>Django</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.7, 3.8, 3.9</td>
<td>2.2, 3.0, 3.1, 3.2, 4.0</td>
</tr>
<tr>
<td>2</td>
<td>3.10</td>
<td>3.2, 4.0</td>
</tr>
</tbody>
</table>
Code and contribution

The code is open source and released under the ISC License (ISCL). It is available on Gitlab and follows the guidelines about Semantic Versioning for transparency within the release cycle and backward compatibility whenever possible.

All contributions are welcome, whether bug reports, code contributions and reviews, documentation or feature requests.
CHAPTER 6

6.1 Getting started

6.1.1 Installation

1. First install the package using pip:

```bash
pip install django-bootstrap-ui
```

2. Add `bootstrap_ui` to your `INSTALLED_APPS` setting:

```python
INSTALLED_APPS = (
    ...
    'bootstrap_ui',
    ...
)
```

6.1.2 Usage

Template skeletons

Prepare your page for HTML5, Bootstrap 3 or 4 using our base templates and provide your content:

1. Extend `html5-skeleton.html`, `bootstrap3-skeleton.html` or `bootstrap4-skeleton.html` in your base template:

```html
{% extends "bootstrap_ui/bootstrap3-skeleton.html" %}
```

2. Fill predefined blocks with your content:
{% block body-content %}
  <h1>Hello, I'm using django-bootstrap-ui!</h1>
{% endblock %}

Bootstrap 3 themes

You can style your page with Bootstrap or Bootswatch themes. Set DJANGO_BOOTSTRAP_UI_THEME to a valid identifier in your settings.py for a project-wide default theme:

```python
# django-bootstrap-ui settings
DJANGO_BOOTSTRAP_UI_THEME = 'bootswatch-paper'
```

Continue reading in our detailed documentation at readthedocs.org.

6.2 Templates

django-bootstrap-ui enables you to quickly prepare your page for Bootstrap and provide your content through a set of predefined templates.

6.2.1 HTML5 skeleton

This is our root template (html5-skeleton.html) with a minimalistic and clean HTML5 skeleton. It comes with a well-thought-out block structure as follows, but is not meant to be extended directly. Extend the Bootstrap 3 skeleton instead.

**head-meta**

Provide additional meta tags within the `<head>`. Example:

```html
{% block head-meta %}
  <meta name="keywords" content="wood, furniture, garden, garden-table, etc.">
{% endblock %}
```

**head-title**

Provide your content for `<title>`. Example:

```html
{% block head-title %}
  Your title
{% endblock %}
```

**head-css**

Provide additional `<link>` tags with css references. Example:

```html
{% block head-css %}
  <link href="css/your-theme.css" media="screen" rel="stylesheet" type="text/css">
{% endblock %}
```
head-javascript

Provide additional `<script>` tags with javascript source references or inline code. Example:

```html
{% block head-javascript %}
  <script src="js/your-script.js" type="text/javascript"></script>
{% endblock %}
```

head-extension

Provide other content within the `<head>`. Example:

```html
{% block head-extension %}
  Whatever content you need at the end of the head.
{% endblock %}
```

body-content

All your page content in `<body>`. Example:

```html
{% block body-content %}
  <h1>Hello, django-bootstrap-ui!</h1>
{% endblock %}
```

body-javascript

Provide additional `<script>` tags with javascript source references or inline code just before `<body>` ends. Example:

```html
{% block body-javascript %}
  <script src="js/your-lately-embedded-script.js" type="text/javascript"></script>
{% endblock %}
```

6.2.2 Bootstrap 4 skeleton

This template (`bootstrap4-skeleton.html`) extends our HTML5 skeleton and provides Bootstrap 4 support. Extend this to get a working, bootstraped page layout.

```html
{% extends "bootstrap_ui/bootstrap4-skeleton.html" %}
```

There are no additional blocks besides the above, the following ones are sensibly used by this template:

- head-meta
- head-css
- head-javascript
- body-javascript

Be careful when you are going to use these blocks with your own stuff. Remember to apply `{{ block.super }}` so you won’t overwrite existing and necessary Bootstrap resources. Example:
6.2.3 Bootstrap 3 skeleton

This template (bootstrap3-skeleton.html) extends our HTML5 skeleton and provides Bootstrap 3 support. Extend this to get a working, bootstrapped page layout.

There are no additional blocks besides the above, the following ones are sensibly used by this template:

- head-meta
- head-css
- head-javascript
- body-javascript

Be careful when you are going to use these blocks with your own stuff. Remember to apply \{ block.super \} so you won’t overwrite existing and necessary Bootstrap resources. Example:

Apply themes

By default the standard, plain Bootstrap style is applied. But in addition, this template is ready to include Bootstrap and Bootswatch themes. You can change this behavior in two ways, they may be used together:

1. Set DJANGO_BOOTSTRAP_UI_THEME to a valid identifier in your settings.py for a project-wide default theme:

   ```python
   # django-bootstrap-ui settings
   DJANGO_BOOTSTRAP_UI_THEME = 'bootswatch-paper'
   ``

2. Set DJANGO_BOOTSTRAP_UI_THEME to a valid identifier in your user’s session or cookie using the set_theme view.

The following table lists currently available themes:
<table>
<thead>
<tr>
<th>Name</th>
<th>Identifier</th>
<th>Provided by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example theme</td>
<td>bootstrap</td>
<td>Bootstrap</td>
</tr>
<tr>
<td>Cerulean</td>
<td>bootswatch-cerulean</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>Cosmo</td>
<td>bootswatch-cosmo</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>Cyborg</td>
<td>bootswatch-cyborg</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>Darkly</td>
<td>bootswatch-darkly</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>Flatly</td>
<td>bootswatch-flatly</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>Journal</td>
<td>bootswatch-journal</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>Lumen</td>
<td>bootswatch-lumen</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>Paper</td>
<td>bootswatch-paper</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>Readable</td>
<td>bootswatch-readable</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>Sandstone</td>
<td>bootswatch-sandstone</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>Simplex</td>
<td>bootswatch-simplex</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>Slate</td>
<td>bootswatch-slate</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>Spacelab</td>
<td>bootswatch-spacelab</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>Superhero</td>
<td>bootswatch-superhero</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>United</td>
<td>bootswatch-united</td>
<td>Bootswatch</td>
</tr>
<tr>
<td>Yeti</td>
<td>bootswatch-yeti</td>
<td>Bootswatch</td>
</tr>
</tbody>
</table>

All other values will be ignored and default to the standard, plain Bootstrap style.

### 6.3 Template tag API

Some utilities and helper tags to make your and our lives easier.

#### 6.3.1 get_value_from_session_or_cookie

Returns a value for a given key from a user’s session or cookie with fallback to global settings:

1. Check `request.session` for `key`
2. Check `request.COOKIES` for `key`
3. Check `settings.py` for `key`
4. Fall back to ''

This tag is implemented as an assignment tag. Usage:

```latex
{% load bootstrap_ui_tags %}
{% get_value_from_session_or_cookie 'DJANGO_BOOTSTRAP_UI_THEME' as theme %}
...
```

### 6.4 Views

django-bootstrap-ui comes with predefined views and corresponding urls that get you right on the track.
6.4.1 Setting the bootstrap theme

The set_theme view

Use the set_theme view to change the style of the page on a per-user basis.

1. determines the page to redirect to after execution in the following manner:
   1. checks the next parameter from request.POST for a valid url
   2. checks the next parameter from request.GET for a valid url
   3. checks the HTTP_REFERER value in request.META for a valid url
   4. falls back to /

2. sets the theme using the theme parameter from request.POST as DJANGO_BOOTSTRAP_UI_THEME in:
   1. the user's session if available
   2. the user's cookie
   3. redirects to the evaluated page

Since this view changes how the user will see the rest of the site, it must only be accessed as a POST request. If called as a GET request, it will redirect to the page in the request (the next parameter) without changing any state.

The set_theme url

A named url set_theme to the above view is available within our urls.py, so you may easily include this route in your project like this:

```python
from django.conf.urls import patterns, include, url

urlpatterns = patterns("",
    ...,
    url(r"^bootstrap_ui/", include("bootstrap_ui.urls")),
    ...
)
```

Then use this url to provide the action attribute for your form:

```html
<form method="post" action="{% url 'set_theme' %}">
  {{ csrf_token }}
  <input type="text" name="next" value="/redirect-to-this-page"/>
  <input type="text" name="theme" value="bootstrap"/>
  <button type="submit">Go!</button>
</form>
```

See Apply themes for valid theme identifiers.

6.5 Release notes

6.5.1 2.0.0
Added

• Add support for Python 3.9 and 3.10
• Add support for Django 3.0, 3.1, 3.2 and 4.0

Changed

• Upgrade Bootstrap 4 template to 4.6.1

Removed

• Remove support for Python 3.6

6.5.2 1.1.0

Added

• Provide full-featured Bootstrap 4 base template
• Integrate Google material design icon font in Bootstrap 4 template

6.5.3 1.0.6

Fixed

• Fixed docs version

6.5.4 1.0.5

Changed

• Improve documentation configuration

6.5.5 1.0.4

Fixed

• Specify master_doc for sphinx on rtd

6.5.6 1.0.3

Changed

• Simplify deploy command
django-bootstrap-ui, Release 2.0

Removed

• Remove auto-documentation of tag reference

6.5.7 1.0.2

Fixed

• Include necessary package data

6.5.8 1.0.1

Fixed

• Find namespaced packages

6.5.9 1.0.0

Added

• Add support for Python 3.7 and 3.8
• Add support for Django 2.2
• Initialise continuous integration with gitlab
• Build distribution after success in test stage
• Deploy build artifacts on tagged commits to pypi

Changed

• Rename bootstrap 3 template for clarification
• Move project to gitlab

Removed

• Remove support for Python 2.7, 3.3, 3.4 and 3.5
• Remove support for Django 1.8, 1.9, 1.10 and 1.11
• Remove bootstrap template tag api
• Remove bootstrap component templates

6.5.10 0.5.1

Changed

• Re-packaged for PyPI deployment
6.5.11 0.5.0

Added

• Added support for Django 1.11
• Added support for Python 3.6

6.5.12 0.4.0

Added

• Added support for Django 1.10

Changed

• Upgraded to Bootstrap 3.3.7
• Upgraded to jQuery 1.12.4
• Upgraded to Font Awesome 4.7.0

Removed

• Dropped support for Django 1.7

6.5.13 0.3.0

Changed

• Upgraded to Bootstrap 3.3.6
• Upgraded to Font Awesome 4.5.0
• Moved project to texperience

6.5.14 0.2.0

Added

• Implemented grid column offsets
• Added grid column pushes and pulls
• Added support for Django 1.9
• Added support for Python 3.5
6.5.15 0.1.0

Added

- Implemented session-based theme switcher
- Integrated Bootstrap and Bootswatch themes
- Provided template for list groups
- Implemented panels in template tag API
- Implemented list groups in template tag API
- Implemented grid system in template tag API
- Introduced template tag API
- Provided full-featured Bootstrap base template
- Provided clean HTML5 base template
- Integrated Font Awesome 4.3.0
- Implemented codebase on Bootstrap 3.3.5
- Established support for Django 1.7 and 1.8
- Established support for Python 2.7, 3.3 and 3.4