django-bootstrap3 Documentation

Release 9.1.0

Dylan Verheul

Nov 13, 2017

Contents

Contents:

CHAPTER 1

Installation

The preferred way to install `django-bootstrap3` is `pip`:

```
$ pip install django-bootstrap3
```

Alternatively, you can install download or clone this repo and install from its folder with:

```
$ pip install -e .
```

In your project, you should add `django-bootstrap3` to your `requirements.txt`.

Be sure to use `virtualenv` if you develop python projects.

Add to INSTALLED_APPS in your `settings.py`:

```
'bootstrap3',
```

After installation, the *Quickstart* will get you on your way to using `django-bootstrap3`.

Quickstart

After *Installation*, you can use `django-bootstrap3` in your templates.:

Load the `bootstrap3` library and use the `bootstrap_*` tags:

2.1 Example template

```
{# Load the tag library #}
{% load bootstrap3 %}

{# Load CSS and JavaScript #}
{% bootstrap_css %}
{% bootstrap_javascript %}

{# Display django.contrib.messages as Bootstrap alerts #}
{% bootstrap_messages %}

{# Display a form #}
<form action="/url/to/submit/" method="post" class="form">
  {% csrf_token %}
  {% bootstrap_form form %}
  {% buttons %}
    <button type="submit" class="btn btn-primary">
      {% bootstrap_icon "star" %} Submit
    </button>
  {% endbuttons %}
</form>

{# Read the documentation for more information #}
```

2.2 Template tags and filters

Refer to *Template tags and filters* for more information.

2.3 Settings

You can set defaults for `django-bootstrap3` in your settings file. Refer to *Settings* for more information.

2.4 Demo application

The demo application provides a number of useful examples.

https://github.com/dyve/django-bootstrap3/tree/master/demo

Template tags and filters

Note: All the following examples it is understood that you have already loaded the `bootstrap3` template tag library, placing the code below in the beginning that each template that `bootstrap3` template tag library will be used. Read the *Installation* and *Quickstart* sections on how to accomplish this.

3.1 bootstrap_form

bootstrap3.templatetags.bootstrap3.**bootstrap_form**(*args*, ***kwargs*)

> Render a form

> **Tag name**:

```
bootstrap_form
```

> **Parameters**:

>> **form** The form that is to be rendered

>> **exclude** A list of field names (comma separated) that should not be rendered E.g. exclude=subject,bcc

>> **error_types** This controls the types of errors that are rendered above the form. Choices are: "all", "field_errors", "non_field_errors" or "none". This will not affect the display of errors on the fields themselves.

>> Default is "non_field_errors".

>> See *bootstrap_field* for other arguments

> **Usage**:

```
{% bootstrap_form form %}
```

Example:

```
{% bootstrap_form form layout='inline' %}
```

3.2 bootstrap_form_errors

bootstrap3.templatetags.bootstrap3.**bootstrap_form_errors**(*args*, ***kwargs*)
Render form errors

Tag name:

```
bootstrap_form_errors
```

Parameters:

form The form that is to be rendered

error_types Control which type of errors should be rendered.

One of the following values:

- `'all'`
- `'field_errors'`
- `'non_field_errors'`

default `'non_field_errors'`

layout Context value that is available in the template `bootstrap3/form_errors.html` as `layout`.

Usage:

```
{% bootstrap_form_errors form %}
```

Example:

```
{% bootstrap_form_errors form layout='inline' %}
```

3.3 bootstrap_formset

bootstrap3.templatetags.bootstrap3.**bootstrap_formset**(*args*, ***kwargs*)
Render a formset

Tag name:

```
bootstrap_formset
```

Parameters:

formset The formset that is being rendered

See *bootstrap_field* for other arguments

Usage:

```
{% bootstrap_formset formset %}
```

Example:

```
{% bootstrap_formset formset layout='horizontal' %}
```

3.4 bootstrap_formset_errors

bootstrap3.templatetags.bootstrap3.**bootstrap_formset_errors**(*args*, ***kwargs*)
   Render formset errors

   Tag name:

```
bootstrap_formset_errors
```

   Parameters:

   formset The formset that is being rendered

   layout Context value that is available in the template `bootstrap3/form_errors.html` as `layout`.

   Usage:

```
{% bootstrap_formset_errors formset %}
```

   Example:

```
{% bootstrap_formset_errors formset layout='inline' %}
```

3.5 bootstrap_field

bootstrap3.templatetags.bootstrap3.**bootstrap_field**(*args*, ***kwargs*)
   Render a field

   Tag name:

```
bootstrap_field
```

   Parameters:

   field The form field to be rendered

   layout If set to `'horizontal'` then the field and label will be rendered side-by-side, as long as there is no `field_class` set as well.

   form_group_class CSS class of the `div` that wraps the field and label.

   > **default** `'form-group'`

   field_class CSS class of the `div` that wraps the field.

   label_class CSS class of the `label` element. Will always have `control-label` as the last CSS class.

   show_help Show the field's help text, if the field has help text.

> > **default** `True`

show_label Whether the show the label of the field.

> > **default** `True`

exclude A list of field names that should not be rendered

set_required When set to `True` and the field is required then the `required` attribute is set on the rendered field. This only works up to Django 1.8. Higher Django versions handle `required` natively.

> > **default** `True`

set_disabled When set to `True` then the `disabled` attribute is set on the rendered field. This only works up to Django 1.8. Higher Django versions handle `disabled` natively.

> > **default** `False`

size Controls the size of the rendered `div.form-group` through the use of CSS classes.

> One of the following values:
>
> - `'small'`
> - `'medium'`
> - `'large'`

placeholder Set/overwrite the field's placeholder.

label Overwrite the field's label.

horizontal_label_class Class used on the label when the `layout` is set to `horizontal`.

> > **default** `'col-md-3'`. Can be changed in *Settings*

horizontal_field_class Class used on the field when the `layout` is set to `horizontal`.

> > **default** `'col-md-9'`. Can be changed in *Settings*

addon_before Text that should be prepended to the form field. Can also be an icon, e.g. `'<span class="glyphicon glyphicon-calendar"></span>'`

> See the *Bootstrap docs <http://getbootstrap.com/components/#input-groups-basic>* for more examples.

addon_after Text that should be appended to the form field. Can also be an icon, e.g. `'<span class="glyphicon glyphicon-calendar"></span>'`

> See the *Bootstrap docs <http://getbootstrap.com/components/#input-groups-basic>* for more examples.

addon_before_class Class used on the span when `addon_before` is used.

> One of the following values:
>
> - `'input-group-addon'`
> - `'input-group-btn'`
>
> > **default** `input-group-addon`

addon_after_class Class used on the span when `addon_after` is used.

> One of the following values:
>
> - `'input-group-addon'`

- `'input-group-btn'`

    default `input-group-addon`

error_css_class CSS class used when the field has an error

default `'has-error'`. Can be changed *Settings*

required_css_class CSS class used on the `div.form-group` to indicate a field is required

default `''`. Can be changed *Settings*

bound_css_class CSS class used when the field is bound

default `'has-success'`. Can be changed *Settings*

Usage:

```
{% bootstrap_field field %}
```

Example:

```
{% bootstrap_field field show_label=False %}
```

3.6 bootstrap_label

`bootstrap3.templatetags.bootstrap3.`**`bootstrap_label`**(*args*, ***kwargs*)
Render a label

Tag name:

```
bootstrap_label
```

Parameters:

content The label's text

label_for The value that will be in the `for` attribute of the rendered `<label>`

label_class The CSS class for the rendered `<label>`

label_title The value that will be in the `title` attribute of the rendered `<label>`

Usage:

```
{% bootstrap_label content %}
```

Example:

```
{% bootstrap_label "Email address" label_for="exampleInputEmail1" %}
```

3.7 bootstrap_button

`bootstrap3.templatetags.bootstrap3.`**`bootstrap_button`**(*args*, ***kwargs*)
Render a button

Tag name:

```
bootstrap_button
```

Parameters:

> **content** The text to be displayed in the button
>
> **button_type** Optional field defining what type of button this is.
>
> > Accepts one of the following values:
> >
> > - `'submit'`
> > - `'reset'`
> > - `'button'`
> > - `'link'`
>
> **icon** Name of an icon to render in the button's visible content. See *bootstrap_icon* for acceptable values.
>
> **button_class** The class of button to use. If none is given, btn-default will be used.
>
> **extra_classes** Any extra CSS classes that should be added to the button.
>
> **size** Optional field to control the size of the button.
>
> > Accepts one of the following values:
> >
> > - `'xs'`
> > - `'sm'`
> > - `'small'`
> > - `'md'`
> > - `'medium'`
> > - `'lg'`
> > - `'large'`
>
> **href** Render the button as an `<a>` element. The `href` attribute is set with this value.
>
> **name** Value of the `name` attribute of the rendered element.
>
> **value** Value of the `value` attribute of the rendered element.

Usage:

```
{% bootstrap_button content %}
```

Example:

```
{% bootstrap_button "Save" button_type="submit" button_class="btn-primary" %}
```

3.8 bootstrap_icon

`bootstrap3.templatetags.bootstrap3.`**`bootstrap_icon`**(*icon*, ***kwargs*)

> Render an icon

> **Tag name**:

```
bootstrap_icon
```

Parameters:

icon Icon name. See the [Bootstrap docs](#) for all icons.

extra_classes Extra CSS classes to add to the icon HTML

title A title for the icon (HTML title attrivute)

Usage:

```
{% bootstrap_icon icon %}
```

Example:

```
{% bootstrap_icon "star" %}
```

3.9 bootstrap_alert

bootstrap3.templatetags.bootstrap3.**bootstrap_alert**(*content*, *alert_type=u'info'*, *dismissable=True*)

Render an alert

Tag name:

```
bootstrap_alert
```

Parameters:

content HTML content of alert

alert_type

- `'info'`
- `'warning'`
- `'danger'`
- `'success'`

  default `'info'`

dismissable boolean, is alert dismissable

  default `True`

Usage:

```
{% bootstrap_alert content %}
```

Example:

```
{% bootstrap_alert "Something went wrong" alert_type='danger' %}
```

3.10 buttons

`bootstrap3.templatetags.bootstrap3.`**`bootstrap_buttons`**(*parser*, *token*)

    Render buttons for form

    Tag name:

```
buttons
```

    Parameters:

        submit Text for a submit button

        reset Text for a reset button

    Usage:

```
{% buttons %}{% endbuttons %}
```

    Example:

```
{% buttons submit='OK' reset="Cancel" %}{% endbuttons %}
```

3.11 bootstrap_messages

`bootstrap3.templatetags.bootstrap3.`**`bootstrap_messages`**(*context*, **args*, ***kwargs*)

    Show django.contrib.messages Messages in Bootstrap alert containers.

    In order to make the alerts dismissable (with the close button), we have to set the jquery parameter too when using the bootstrap_javascript tag.

    Uses the template `bootstrap3/messages.html`.

    Tag name:

```
bootstrap_messages
```

    Parameters:

        None.

    Usage:

```
{% bootstrap_messages %}
```

    Example:

```
{% bootstrap_javascript jquery=1 %}
{% bootstrap_messages %}
```

3.12 bootstrap_pagination

`bootstrap3.templatetags.bootstrap3.`**`bootstrap_pagination`**(*page*, ***kwargs*)

    Render pagination for a page

    Tag name:

```
bootstrap_pagination
```

Parameters:

> **page** The page of results to show.
>
> **pages_to_show** Number of pages in total
>
> > **default** `11`
>
> **url** URL to navigate to for pagination forward and pagination back.
>
> > **default** `None`
>
> **size** Controls the size of the pagination through CSS. Defaults to being normal sized.
>
> > One of the following:
> >
> > * `'small'`
> >
> > * `'large'`
> >
> > > **default** `None`
>
> **extra** Any extra page parameters.
>
> > **default** `None`
>
> **parameter_name** Name of the paging URL parameter.
>
> > **default** `'page'`

Usage:

```
{% bootstrap_pagination page %}
```

Example:

```
{% bootstrap_pagination lines url="/pagination?page=1" size="large" %}
{% bootstrap_pagination page_obj extra=request.GET.urlencode %}
```

3.13 bootstrap_jquery_url

bootstrap3.templatetags.bootstrap3.**bootstrap_jquery_url**()
   Tag name:

```
bootstrap_jquery_url
```

Return the full url to jQuery file to use

Default value: `//code.jquery.com/jquery.min.js`

This value is configurable, see Settings section

Usage:

```
{% bootstrap_jquery_url %}
```

Example:

```
{% bootstrap_jquery_url %}
```

3.14 bootstrap_javascript_url

bootstrap3.templatetags.bootstrap3.**bootstrap_javascript_url**()
    Return the full url to the Bootstrap JavaScript library

    Default value: `None`

    This value is configurable, see Settings section

    Tag name:

```
bootstrap_javascript_url
```

    Usage:

```
{% bootstrap_javascript_url %}
```

    Example:

```
{% bootstrap_javascript_url %}
```

3.15 bootstrap_css_url

bootstrap3.templatetags.bootstrap3.**bootstrap_css_url**()
    Return the full url to the Bootstrap CSS library

    Default value: `None`

    This value is configurable, see Settings section

    Tag name:

```
bootstrap_css_url
```

    Usage:

```
{% bootstrap_css_url %}
```

    Example:

```
{% bootstrap_css_url %}
```

3.16 bootstrap_css

bootstrap3.templatetags.bootstrap3.**bootstrap_css**()
    Return HTML for Bootstrap CSS. Adjust url in settings. If no url is returned, we don't want this statement to
    return any HTML. This is intended behavior.

    Default value: `None`

    This value is configurable, see Settings section

Tag name:

```
bootstrap_css
```

Usage:

```
{% bootstrap_css %}
```

Example:

```
{% bootstrap_css %}
```

3.17 bootstrap_javascript

bootstrap3.templatetags.bootstrap3.**bootstrap_javascript**(*jquery=None*)
Return HTML for Bootstrap JavaScript.

Adjust url in settings. If no url is returned, we don't want this statement to return any HTML. This is intended behavior.

Default value: None

This value is configurable, see Settings section

Tag name:

```
bootstrap_javascript
```

Parameters:

jquery Truthy to include jQuery as well as Bootstrap

Usage:

```
{% bootstrap_javascript %}
```

Example:

```
{% bootstrap_javascript jquery=1 %}
```

Settings

The django-bootstrap3 has some pre-configured settings.

They can be modified by adding a dict variable called `BOOTSTRAP3` in your `settings.py` and customizing the values you want;

The `BOOTSTRAP3` dict variable contains these settings and defaults:

```python
# Default settings
BOOTSTRAP3 = {

    # The URL to the jQuery JavaScript file
    'jquery_url': '//code.jquery.com/jquery.min.js',

    # The Bootstrap base URL
    'base_url': '//maxcdn.bootstrapcdn.com/bootstrap/3.3.7/',

    # The complete URL to the Bootstrap CSS file (None means derive it from base_url)
    'css_url': None,

    # The complete URL to the Bootstrap CSS file (None means no theme)
    'theme_url': None,

    # The complete URL to the Bootstrap JavaScript file (None means derive it from
    # base_url)
    'javascript_url': None,

    # Put JavaScript in the HEAD section of the HTML document (only relevant if you
    # use bootstrap3.html)
    'javascript_in_head': False,

    # Include jQuery with Bootstrap JavaScript (affects django-bootstrap3 template
    # tags)
    'include_jquery': False,

    # Label class to use in horizontal forms
    'horizontal_label_class': 'col-md-3',
```

```
    # Field class to use in horizontal forms
    'horizontal_field_class': 'col-md-9',

    # Set HTML required attribute on required fields, for Django <= 1.8 only
    'set_required': True,

    # Set HTML disabled attribute on disabled fields, for Django <= 1.8 only
    'set_disabled': False,

    # Set placeholder attributes to label if no placeholder is provided.
    # This also considers the 'label' option of {% bootstrap_field %} tags.
    'set_placeholder': True,

    # Class to indicate required (better to set this in your Django form)
    'required_css_class': '',

    # Class to indicate error (better to set this in your Django form)
    'error_css_class': 'has-error',

    # Class to indicate success, meaning the field has valid input (better to set
→this in your Django form)
    'success_css_class': 'has-success',

    # Renderers (only set these if you have studied the source and understand the
→inner workings)
    'formset_renderers':{
        'default': 'bootstrap3.renderers.FormsetRenderer',
    },
    'form_renderers': {
        'default': 'bootstrap3.renderers.FormRenderer',
    },
    'field_renderers': {
        'default': 'bootstrap3.renderers.FieldRenderer',
        'inline': 'bootstrap3.renderers.InlineFieldRenderer',
    },
}
```

Templates

You can customize the output of `django-bootstrap3` by writing your own templates. These templates are available:

5.1 bootstrap3/field_help_text_and_errors.html

This renders the help text and error of each field.

Variable `help_text_and_errors` contains an array of strings.

5.2 bootstrap3/form_errors.html

This renders the non field errors of a form.

Variable `errors` contains an array of strings.

5.3 bootstrap3/messages.html

This renders the Django messages variable.

Variable `messages` contains the messages as described in https://docs.djangoproject.com/en/dev/ref/contrib/messages/#displaying-messages

`messages` is passed through three built-in filters

safe <https://docs.djangoproject.com/en/dev/ref/templates/builtins/#std:templatefilter-safe>

urlize <https://docs.djangoproject.com/en/dev/ref/templates/builtins/#std:templatefilter-urlize>

linebreaksbr <https://docs.djangoproject.com/en/dev/ref/templates/builtins/#std:templatefilter-linebreaksbr>

5.4 Other

There are two more templates, `bootstrap3/bootstrap3.html` and `bootstrap3/pagination.html`.
You should consider these private for now, meaning you can use them but not modify them.

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

6.1 Types of Contributions

6.1.1 Report Bugs

Report bugs at https://github.com/dyve/django-bootstrap3/issues.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

6.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" is open to whoever wants to implement it.

6.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "feature" is open to whoever wants to implement it.

6.1.4 Write Documentation

django-bootstrap3 could always use more documentation, whether as part of the official django-bootstrap3 docs, in docstrings, or even on the web in blog posts, articles, and such.

6.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/dyve/django-bootstrap3/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

6.2 Get Started!

Ready to contribute? Here's how to set up *django-bootstrap3* for local development.

1. Fork the *django-bootstrap3* repo on GitHub.
2. Clone your fork locally:

```
$ git clone https://github.com/your_username_here/django-bootstrap3.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv django-bootstrap3
$ cd django-bootstrap3/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 bootstrap3 tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

6.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.

2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.

3. The pull request should work for Python 2.6, 2.7, and 3.3, and for PyPy. Check https://travis-ci.org/dyve/django-bootstrap3/pull_requests and make sure that the tests pass for all supported Python versions.

6.4 Tips

To run a subset of tests:

```
$ python -m unittest tests.test_bootstrap3
```

Authors

This application is developed and maintained by Zostera.

7.1 Development Lead

- Dylan Verheul <dylan@dyve.net>

7.2 Contributors

- Allard Stijnman <a.g.stijnman@gmail.com>
- Austin Whittier <austin.whitt@gmail.com>
- Caio Ariede <caio.ariede@gmail.com>
- Fabio C. Barrionuevo da Luz <bnafta@gmail.com>
- Fabio Perfetti <perfabio87@gmail.com>
- Irving Ckam <https://github.com/ickam>
- Jay Pipes <jaypipes@gmail.com>
- Jonas Hagstedt <hagstedt@gmail.com>
- Jordan Starcher <jstarcher@gmail.com>
- Juan Carlos <juancarlospaco@gmail.com>
- Markus Holtermann <info@markusholtermann.eu>
- Martin Koistinen <mkoistinen@gmail.com>
- Nick S <nsmith448@gmail.com>
- Owais Lone <loneowais@gmail.com>

- pmav99 <pmav99@users.noreply.github.com>

- Richard Hajdu <tuskone16@gmail.com>

History

8.1 In Progress

- Switched to master as main branch, deleted other branches
- Switched to twine for publication on PyPI

8.2 9.1.0 (2017-10-27)

- Mention django-bootstrap4 in README
- Rewrite *tox* test matrix to focus on Django releases rather than Python versions
- Add tests for Django master branch (>= 2)
- Add *label* override for *{% bootstrap_field %}*

8.3 9.0.0 (2017-07-11)

- Renamed requirements-dev.txt back to requirements.txt because that suits ReadTheDocs better
- Added *error_types* support on bootstrap3_form (thanks @mkoistinen and @ickam)
- **BREAKING** Default setting of *error_types* to *non_field_errors* is different fro behavior in versions < 9

8.4 8.2.3 (2017-05-05)

- Renamed requirements.txt to requirements-dev.txt
- Tweaks to tests and CI (see #400)

- Prepared test for geometry fields (disabled, blocked by Django update, see #392)
- Bug fixes for add ons and placeholders (thanks @jaimesanz, @cybojenix and @marc-gist)
- Improve documentation for pagination with GET parameters (thanks @nspo)
- Add unicode test for help_text
- Removed tests for Python 3.2 from tox and Travis CI (no longer supported by Django 1.8)

8.5 8.2.2 (2017-04-03)

- Fix invalid HTML in help texts (thanks @luksen)
- Added *mark_safe* to placeholder (thanks @ppo)
- Fix DateWidget import for newer Django versions (thanks @clokep)

8.6 8.2.1 (2017-02-23)

- Support for local languages in *url_replace_param* on Python 2 (#362, thanks @aamalev)
- Correct checking Mapping instance (#363, thanks @aamalev)
- Fix Django 1.11 import bug (see #369)
- Add Django 1.11 and Python 3.6 to tests
- Fix sdist issue with .pyc files

8.7 8.1.0 (2017-01-12)

- Rolled back subresource integrity (see #353)
- Documentation fix (thanks @clokep)

8.8 8.0.0 (2017-01-06)

- **BREAKING** For Django >= 1.10 Remove everything to do with setting HTML attributes *required* (#337) and *disabled* (#345)
- Add *id* parameter to bootstrap_button (#214)
- Add *set_placeholder* to field and form renderers (#339, thanks @predatell)
- Default button type to *btn-default*
- Add *addon_before_class* and *addon_after_class* (#295, thanks @DanWright91 and others)
- Fix handling of error class (#170)
- No size class for checkboxes (#318, thanks @cybojenix)
- Fix warnings during install (thanks @mfcovington)
- Fix rare RunTimeError when working without database (#346, thanks @Mactory)

- Add subresource integrity to external components (thanks @mfcovington and @Alex131089)

- Several improvements to documentation, tests, and comments. Thanks all!

8.9 7.1.0 (2016-09-16)

- Print help text and errors in their own block (#329, thanks @Matoking)

- Improved page urls in pagination (fixes #323)

- Changed setup.py to allow *setup.py test* run tests

- Removed link target from active page in pagination (fixes #328)

- Fixed example for bootstrap_label (fixed #332)

- Fixed tests to support Django 1.10 handling of required attribute, see #337 (needs fixing)

- Added tests for Django 1.10

- Bootstrap to 3.3.7

8.10 7.0.1 (2016-03-23)

- Fixed bug with widget attrs consistency (@onysos)

8.11 7.0.0 (2016-02-24)

- Dropped support for Django < 1.8

- Dropped support for Python < 2.7

- Fix page number bug (thanks @frewsxcv)

- Fix template context warning (thanks @jieter and @jonashaag)

- Update to Bootstrap 3.3.6 (@nikolas)

- Show links and newlines in messages (@jakub3279)

- CSS classes arguments passed to the bootstrap_form are now working (@gordon)

- Support for Django 1.9/Python 3.5 (@jieter and @jonashaag)

- Better Travis CI Django versions (thanks @jonashaag)

- Improved handling of messages in *bootstrap_messages* (thanks @frewsxcv and @rjsparks)

8.12 6.2.2 (2015-08-20)

- Bug fix for escaped icons in buttons (reported by @jlec)

8.13 6.2.1 (2015-08-19)

- Bug fix for whitespace in label placeholders (@Grelek)

8.14 6.2.0 (2015-08-15)

- Improved tests
- Make simple_tag output safe in Django 1.9
- Better support for MultiWidgets (@xrmx)
- Better documentation (@Moustacha)

8.15 6.1.0 (2015-06-25)

- Upgrade to Bootstrap 3.3.5
- Properly quote help text (@joshkel)

8.16 6.0.0 (2015-04-21)

- No more media="screen" in CSS tags, complying to Bootstraps examples

8.17 5.4.0 (2015-04-21)

- No more forcing btn-primary when another button class is specified (@takuchanno2)
- Added value option to buttons (@TyVik)
- Switched CDN to //maxcdn.bootstrapcdn.com/bootstrap/3.3.4/ (@djangoic)

8.18 5.3.1 (2015-04-08)

- Fix Django 1.8 importlib warnings
- Set defaults for horizontal-form to col-md-3 for label, col-md-9 for field
- Various bug fixes
- Fix version number typo

8.19 5.2.0 (2015-03-25)

- Upgrade to Bootstrap 3.3.4
- Fix required bug for checkboxes
- Various bug fixes

8.20 5.1.1 (2015-01-22)

- Fix checkbox display bug

8.21 5.1.0 (2015-01-22)

- Make Bootstrap 3.3.2 default
- Fix issue #140 (bad behaviour in Python 3)

8.22 5.0.3 (2014-12-02)

- Fixing tests for older Django and Python versions

8.23 5.0.2 (2014-11-24)

- Cleaning up some mess in 5.0.1 created by PyPI malfunction

8.24 5.0.1 (2014-11-21)

- Bug fixes and update to Bootstrap 3.3.1

8.25 4.11.0 (2014-08-19)

- Improved handling and control of form classes for error and success

8.26 4.10.1 (2014-08-18)

- Bug fixes, test fixes, documentation fixes

8.27 4.10.0 (2014-08-12)

- Template tag *bootstrap_icon* now supports a *title* parameter

8.28 4.9.2 (2014-08-11)

- Fixed bug causing problems with setting classes for horizontal forms

8.29 4.9.1 (2014-08-10)

- Fixed test for Django 1.4

8.30 4.9.0 (2014-08-09)

- New parameter *href* for *bootstrap_button*, if provided will render *a* tag instead of *button* tag

8.31 4.8.2 (2014-07-10)

- Internal fixes to master branch

8.32 4.8.1 (2014-07-10)

- Make extra classes override bootstrap defaults

8.33 4.8.0 (2014-07-10)

- Introduced new setting *set_placeholder*, default True

8.34 4.7.1 (2014-07-07)

- Fixed rendering of various sizes (as introduced in 4.7.0)
- Upgrade to Bootstrap 3.2.0 as default version

8.35 4.7.0 (2014-06-04)

- *size* option added to formsets, forms, fields and buttons

8.36 4.6.0 (2014-05-22)

- new *bootstrap_formset_errors* tag

8.37 4.5.0 (2014-05-21)

- bug fixes in formsets
- new formset renderer
- new *bootstrap_form_errors* tag

8.38 4.4.2 (2014-05-20)

- documentation now mentions templates

8.39 4.4.1 (2014-05-08)

- bug fixes
- documentation fixes
- test coverage on coveralls.io

8.40 4.4.0 (2014-05-01)

- added *bootstrap_alert* template tag

8.41 4.3.0 (2014-04-25)

- added *required_css_class* and *error_css_class* as optional settings (global) and parameters (form and field rendering)

8.42 4.2.0 (2014-04-06)

- moved styling of form level errors to template
- bug fixes

8.43 4.1.1 (2014-04-06)

- moved all text conversions to text_value

8.44 4.1.0 (2014-04-05)

- typo fix and internal branching changes

8.45 4.0.3 (2014-04-03)

- fixed checkbox label bug in vertical and inline forms

8.46 4.0.2 (2014-04-02)

- fixed bug in vertical form rendering

8.47 4.0.1 (2014-03-29)

- fixed unicode bug and added unicode label to tests

8.48 4.0.0 (2014-03-28)

- use renderer classes for generating HTML
- several bug fixes

8.49 3.3.0 (2014-03-19)

- use Django forms css classes for indicating required and error on fields

8.50 3.2.1 (2014-03-16)

- improved form rendering

8.51 3.2.0 (2014-03-11)

- support for addons

8.52 3.1.0 (2014-03-03)

- improve compatibility with Django < 1.5

8.53 3.0.0 (2014-02-28)

- added support for themes (fix issue #74)
- show inline form errors in field title (fix issue #81)
- fixed bugs in demo application
- update to newest Bootstrap (fix issue #83)

8.54 2.6.0 (2014-02-20)

- new setting *set_required* to control setting of HTML *required* attribute (fix issue #76)

8.55 2.5.6 (2014-01-23)

- project refactored

- added skeleton for creating documentation (fix issue #30)

- fixed *FileField* issues

Index

B