Django AdminLTE 2 Documentation

Release 0.1

Adam Charnock

Jul 02, 2018

Contents

1 Contents **3**
- 1.1 Quickstart . . . 3
- 1.2 Templates & Blocks Reference . . . 4
- 1.3 Templates & Blocks Reference . . . 15

2 Credits **17**

3 Indices and tables **19**

Django AdminLTE2 provides extensible templates for making use of the general purpose AdminLTE2 theme.

CHAPTER 1

Contents

1.1 Quickstart

Install the pip package:

```
pip install django-adminlte2
```

Add the apps to `INSTALLED_APPS`:

```
INSTALLED_APPS = [
    # Any apps which will override adminlte's templates (i.e. your apps)
    ...

    # The general purpose templates
    'django_adminlte',

    # Optional: Skin for the admin interface
    'django_adminlte_theme',

    # Any apps which need to have their templates overridden by adminlte
    'django.contrib.admin',
    ...
]
```

Important: Take note of the ordering of `INSTALLED_APPS`. If you find templates are not being found & used as expected it is probably due to a problem here.

Django looks for a template in the order in which apps are listed in `INSTALLED_APPS`, hence the ordering above.

If you only wish to skin the admin interface, you can stop here. Your admin interface should now be displaying with the AdminLTE theme.

You can also make use of the AdminLTE theme for your app. This may be particularly useful for internal (non-public) apps which need a quick and effective layout.

1.1.1 Using the templates in your app

The base AdminLTE template provides much of what you need, but you'll need to customise it in some ways to meet your needs. In particular, no navigation is provided (we'll cover this shortly).

To add & modify the functionality of the base template you should create your own base template. This template should extend `adminlte/base.html`. Several blocks are available for you to extend.

```
{% extends 'adminlte/base.html' %}

{% block title %}My App{% endblock %}
{% block content %}
    Just some example content
{% endblock %}
```

Take a look at the base template to see the available blocks.

1.1.2 Adding navigation

The base template includes a number of other templates in order to create the whole.

For example, the sidebar navigation resides in `adminlte/lib/_main_sidebar.html`. To customise this template you should create a template of the same path & name in your app's templates folder.

Important: Your app(s) must be listed before `django_adminlte2` in `INSTALLED_APPS`. Otherwise Django will find the default default provided by `django-adminlte2` before your customised one.

In **your own app** create the template `adminlte/lib/_main_sidebar.html`:

```
{% extends 'adminlte/lib/_main_sidebar.html' %}

{% block nav_links %}
    <li>
        <a href="/some/url">
            <i class="fa fa-dashboard"></i> <span>Home</span>
        </a>
    </li>
    <li>
        <a href="/some/url">
            <i class="fa fa-user"></i> <span>Users</span>
        </a>
    </li>
{% endblock nav_links %}
```

You should now find the navigation has updated.

1.2 Templates & Blocks Reference

Contents

- *Templates & Blocks Reference*
  - *Base templates*
    - *adminlte/base.html*
      - *Commonly used blocks*
      - *Block reference*
      - `title_outer`
      - `title`
      - `meta`
      - `stylesheets`
      - `extra_head`
      - `body_class`
      - `nav_header`
      - `nav_sidebar`
      - `content_wrapper`
      - `content_header`
      - `page_name`
      - `page_description`
      - `no_description`
      - `page_actions`
      - `breadcrumbs`
      - `content_outer`
      - `messages`
      - `content_block_wrap`
      - `content`
      - `nav_footer`
      - `javascript`
      - `extra_foot`
      - `body`
    - *adminlte/login.html*
      - *Block reference*
      - `logo`
      - `logo_text`
      - `logo_href`
      - `login_form`

- login_form_links
- *Include templates*
  - *adminlte/lib/_main_sidebar.html*
    - *Block Reference*
    - user_panel
    - form
    - nav_links
    - nav_links_ul
    - nav_links_outer
  - *adminlte/lib/_main_header.html*
    - *Block Reference*
    - logo
    - logo_href
    - logo_text
    - logo_text_small
    - nav_bar
    - nav_bar_center
    - header_dropdowns
    - user_header
    - menu_footer
    - change_password_url
    - logout_url
  - *adminlte/lib/_main_footer.html*
    - *Block Reference*
    - footer_right
    - version
    - footer_left
    - legal

Below you will find reference for each available template, and the blocks that can be overridden.

Note than when overriding some blocks it may make sense to make use of `{{ block.super }}` (inheritance docs).

1.2.1 Base templates

adminlte/base.html

The primary base template which provides a sidebar, top navigation with user information, and footer.

Commonly used blocks

- *title*
- *page_name*
- *page_description*
- *content*
- *stylesheets*
- *javascript*

Block reference

title_outer

Wrapper around the the outside of the <title> tag. Default:

```
{% block title_outer %}
    <title>{% block title %}{{ site.name }}{% endblock %}</title>
{% endblock %}
```

title

Contents of the pages <title> tag. Default:

```
<title>{% block title %}{{ site.name }}{% endblock %}</title>
```

This will set a sensible default if the site variable references the current site object (from Django's sites framework)

meta

All <meta> tags which appear in the pages <head>. Default:

```
{% block meta %}
    <meta charset="utf-8">
    <meta http-equiv="X-UA-COMPATIBLE" content="IE=edge">
    <meta content="width=device-width, initial-scale=1, maximum-scale=1,␣
↪user-scalable=no" name="viewport">
{% endblock %}
```

Consider making use of {{ block.super }} when overriding this block.

stylesheets

All <style> tags which appear in the pages <head>. By default this includes all content from adminlte/lib/_styles.html.

Consider making use of {{ block.super }} when overriding this block.

extra_head

Additional HTML to be placed before the </head> tag. Empty by default.

body_class

Additional CSS classes which can be placed into the <body> tag's class attribute.

nav_header

Wrapper around the entirety of the main header. Default:

```
{% block nav_header %}
    {% include 'adminlte/lib/_main_header.html' %}
{% endblock %}
```

If you wish to customise the entirety of the main header you may override this block, otherwise you can redefine adminlte/lib/_main_header.html to get finer control over your changes.

nav_sidebar

Wrapper around the entirety of the sidebar. Default:

```
{% block nav_sidebar %}
    {% include 'adminlte/lib/_main_sidebar.html' %}
{% endblock %}
```

If you wish to customise the entirety of the sidebar you may override this block, otherwise you can redefine adminlte/lib/_main_header.html to get finer control over your changes.

content_wrapper

Wrapper around all of the content area (including the content header, messages, and actual page content).

You probably want to override the ‘‘content‘‘ block instead.

content_header

The header that appears over the page content, but within the content area of the design.

Default:

```
{% block content_header %}
    <section class="content-header">
        <h1>
            {% block page_name %}{% endblock %}
            {% block no_description %}
            <small>{% block page_description %}{% endblock %}</small>
            {% endblock %}
        </h1>
        {% block breadcrumbs %}
```

(continues on next page)

(continued from previous page)

```
            {# Breadcrumb implementation left to developers #}
        {% endblock %}
    </section>
{% endblock %}
```

page_name

The name of the page as will be displayed in the content header.

page_description

The description of the page tht will appear alongside the page name in the header.

no_description

If no description is to be displayed, you can implement this as an empty block to remove the wrapper HTML. For example:

```
{% block no_description %}{% endblock %}
```

page_actions

Generally used to display actions/buttons relevant to the current page. For example:

```
{% block page_actions %}
    <a href="{% url 'alerts:create' %}" class="btn btn-success btn-sm">
↪Create new</a>
{% endblock %}
```

breadcrumbs

Use the block the implement your breadcrumbs if desired.

Todo: Provide tools to make the generation of breadcrumbs easier

---

content_outer

Wraps the outside of the content area and any messages.

You probably want to override the ‘‘content‘‘ block instead.

messages

Wrapper around the entirety of the message area. Default:

```
{% block messages %}
    {% include 'adminlte/lib/_messages.html' %}
{% endblock %}
```

See the Django messages framework.

content_block_wrap

Wraps the `content` block. May be useful in some cases.

content

Block for the main content which will be displayed in the page. Empty by default.

nav_footer

Wrapper around the entirety of the main footer. Default:

```
{% block nav_footer %}
    {% include 'adminlte/lib/_main_footer.html' %}
{% endblock %}
```

If you wish to customise the entirety of the main footer you may override this block, otherwise you can redefine `adminlte/lib/_main_footer.html` to get finer control over your changes.

javascript

All `<script>` tags which appear before the `</body>` tag. By default this includes all content from `adminlte/lib/_scripts.html`.

Consider making use of `{{ block.super }}` when overriding this block.

extra_foot

Additional HTML to be placed before the `</body>` tag. Empty by default.

body

Wraps the entire contents of the body tag, excluding the `javascript` and `extra_foot` blocks. Define if you wish to replace the entire body of the page

adminlte/login.html

Base template for a login interface. This excludes the navigational elements which are usually present. Example:

```
{% extends "adminlte/login.html" %}

{% block form %}
    <form method="post">
        {% csrf_token %}
        {{ form  }}
    </form>
{% endblock %}
```

Block reference

The login form defines some blocks in addition to those available on *adminlte/base.html*.

logo

Wraps the logo section of the login page. Default:

```
{% block logo %}
<div class="login-logo">
    <a href="{% block logo_href %}/{% endblock %}">{% block logo_text %}<b>
↪Admin</b>LTE{% endblock %}</a>
</div>
{% endblock %}
```

logo_text

The name of the site as shown above the login form. Default:

```
{% block logo_text %}<b>Admin</b>LTE{% endblock %}
```

logo_href

URL the logo should link to. Default: /

login_form

The form to be displayed. Defaults to a static HTML form.

login_form_links

Show to the left of the login button. A useful place for a forgotten password link.

1.2.2 Include templates

Much of the HTML rendering is done in included template files. These files reside in `adminlte/lib/`.

The easiest way to do this is to create a file of the same path and name in your app's templates folder. This new template can then extend the original template and tweak blocks as necessary (or, if you wish, forgo the extending the reimplement the entire template).

Here is an example of the overriding and extension. We will be overriding the sidebar template (`adminlte/lib/_main_sidebar.html`), so we'll create a template called `my_app_name/templates/adminlte/lib/_main_sidebar.html`:

```
{% extends 'adminlte/lib/_main_sidebar.html' %}

{% block nav_links %}
    <li>
        <a href="/some/url">
            <i class="fa fa-dashboard"></i> <span>Home</span>
        </a>
    </li>
    <li>
        <a href="/some/url">
            <i class="fa fa-user"></i> <span>Users</span>
        </a>
    </li>
{% endblock nav_links %}
```

adminlte/lib/_main_sidebar.html

Renders the sidebar navigation. You'll likely need to implement this template at a minimum.

Block Reference

`user_panel`

Wraps the user details panel

`form`

An empty tag where you may wish to include a form. The AdminLTE examples place a search box here.

`nav_links`

Renders the `<li>` elements for the navigation. See above for an example.

`nav_links_ul`

Wrapper around the entire `<ul>` element containing the navigation.

You probably want to use *nav_links*.

nav_links_outer

Wrapper within the <ul> element around all <li> elements.

You probably want to use *nav_links*.

adminlte/lib/_main_header.html

Renders the header. Contains the site name and details regarding the currently logged in user.

Block Reference

logo

Wraps the logo HTML. Default:

```
{% block logo %}
<a href="{% block logo_href %}/{% endblock %}" class="logo">
    <!-- mini logo for sidebar mini 50x50 pixels -->
    <span class="logo-mini"><b>On</b>ly</span>
    <!-- logo for regular state and mobile devices -->
    <span class="logo-lg"><b>Only</b>Admin</span>
</a>
{% endblock %}
```

logo_href

URL the logo should link to. Default: /

logo_text

The name of the site as shown in the header. Default:

```
{% block logo_text %}<b>Admin</b>LTE{% endblock %}
```

logo_text_small

The logo name of the site as show in the header (used on narrow/mobile screens). Default:

```
{% block logo_text_small %}<b>A</b>LTE{% endblock %}
```

nav_bar

The entirety of the header navigation

nav_bar_center

An empty block in the center of the main nav bar.

header_dropdowns

The dropdown menus in the header.

user_header

The contents of the user dropdown in the header. Default:

```
{% block user_header %}
<li class="user-header">
    <img src="{% avatar_url size=180 %}" class="img-circle" alt="User Image">
    <p>
        {% firstof request.user.get_short_name request.user.get_username %}
        <small>Member since {{ request.user.date_joined }}</small>
    </p>
</li>
{% endblock %}
```

menu_footer

The footer of the user dropdown. Normally used for actions such as 'Change password' and 'logout'. Default:

```
{% block menu_footer %}
<li class="user-footer">
    <div class="pull-left">
        <a href="{% block change_password_url %}{% url 'admin:password_change
↪' %}{% endblock %}"
           class="btn btn-default btn-flat">{% trans 'Change password' %}</a>
    </div>
    <div class="pull-right">
        <a href="{% block logout_url %}{% logout_url %}{% endblock %}" class=
↪"btn btn-default btn-flat">Sign out</a>
    </div>
</li>
{% endblock %}
```

change_password_url

The URL to the change password interface (defaults to Django admin's change password page)

logout_url

The URL used for logging out the current user. Defaults to the value given in the `LOGOUT_URL` setting, or `/logout` if not set.

adminlte/lib/_main_footer.html

Renders the footer containing (by default) a legal notice and software version.

Default footer content:

```
<footer class="main-footer">
    <div class="pull-right hidden-xs">
        {% block footer_right %}
            <b>Version</b> {% block version %}0.1{% endblock %}
        {% endblock %}
    </div>

    {% block footer_left %}
    {% block legal %}
    <strong>Copyright &copy; {% now "Y" %}{% if not site %}.{% endif %}
        {% if site %}
            <a href="http://{{ site.domain }}">{{ site.name }}</a>
        {% endif %}
    </strong> All rights
    reserved.
    {% endblock %}
    {% endblock %}
</footer>
```

Block Reference

`footer_right`

Content to be displayed on the right of the footer. See above for default.

`version`

The current version of the software. Shown in `footer_right` by default.

`footer_left`

The left hand content of the footer. Contains only `legal` by default.

`legal`

Legal notice. Will include a copyright notice referencing the current date and site name (if present).

1.3 Templates & Blocks Reference

Contents

- *Templates & Blocks Reference*

```
        - {% add_active %}
        - {% avatar_url %}
        - {% logout_url %}
```

Some utility template tags are provided which you may find useful.

1.3.1 {% add_active %}

Returns the string " active " if the current url matches the one provided. You may find this useful in your lib/_main_sidebar.html template.

Example:

```
<li class="{% add_active 'myapp:detail' object.pk %}">
    <a href="{% url 'myapp:detail' object.pk %}">Details</a>
</li>
```

This will normally match any URL which starts with the provided value. Therefore the 'active' class will be applied for child pages too.

You can also specify the exact_only parameter which will override this behaviour. This is often useful for the home page:

```
<li class="{% add_active 'myapp:home' exact_match=True %}">
    <a href="{% url 'myapp:home' %}">Home</a>
</li>
```

1.3.2 {% avatar_url %}

Used in the based templates to find an avatar for the current user (uses Gravatar).

1.3.3 {% logout_url %}

Used in the based templates to determine the default logout url by looking for the LOGOUT_URL setting.

CHAPTER 2

Credits

This project a based heavily on work by the following:

- *dnaextrim* for django_adminlte_x
- *beastbikes* for django-adminlte

CHAPTER 3

Indices and tables

- genindex
- modindex
- search