
Flask-Styleguide Documentation

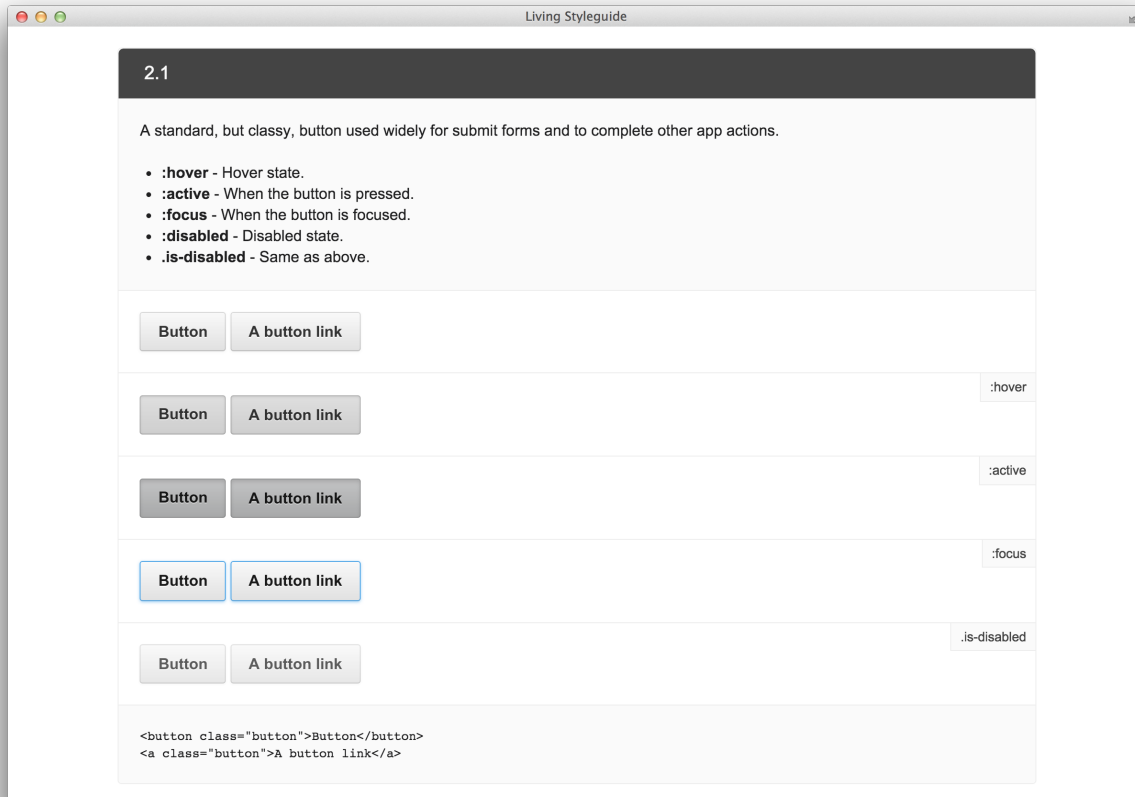
Release 0.4.0

Vital Kudzelka

November 12, 2015

1	Getting started	3
2	Contributing	5
3	Changelog	7
3.1	0.4.0	7
3.2	0.3.0	7
3.3	0.2.0	7
3.4	0.1.5	7
3.5	0.1.4	7
3.6	0.1.3	8
3.7	0.1.2	8
3.8	0.1.1	8
3.9	0.1.0	8

Extension provides an easy way to automatically generate living styleguide for your application from [KSS documentation](#) format.



Note: KSS is a documentation for humans. It's human readable, machine parsable, and easy to remember. [Learn the syntax in less than 5 minutes.](#)

```
// A standard, but classy, button used widely for submit forms and
// to complete other app actions.
//
// :hover - Hover state.
// :active - When the button is pressed.
// :focus - When the button is focused.
// :disabled - Disabled state.
// .is-disabled - Same as above.
//
// Styleguide 2.1.
```

Getting started

Use default Flask API to initialize application and bound it to application object:

```
from flask import Flask
from flask_styleguide import Styleguide

app = Flask(__name__)
styleguide = Styleguide(app)
```

Define endpoint for your styleguide in application or blueprint:

```
@app.route('/styleguide/')
def styleguide():
    return render_template('styleguide.html')
```

The new jinja tag `styleguide` becomes available when extension is initialized, so it's easy to scaffold your styleguide:

```
{% extends 'layout.html' %}
{% block main %}
    {% styleguide "2.1" %}
        <button class="button$modifier_class">Button</button>
        <a class="button$modifier_class">A button link</a>
    {% endstyleguide %}
{% endblock %}
```

Note: The `styleguide/section.html` template will be used for each section in your styleguide (like 2.1 in the example above). You are free to define your own template or reuse this one:

```
<article class="style-guide" id="{{ section.section }}">
    <h4 class="style-guide-reference">{{ section.section }}</h4>

    <div class="style-guide-description">
        <p>{{ section.description|safe }}</p>
        {% if section.modifiers %}
            <ul class="style-guide-modifiers">
                {% for m in section.modifiers %}
                    <li><strong>{{ m.name }}</strong> - {{ m.description }}</li>
                {% endfor %}
            </ul>
        {% endif %}
    </div>

    <div class="style-guide-element">
```

```
    {{ section.example|safe }}
</div>

{% for m in section.modifiers %}
    <div class="style-guide-element">
        <small class="style-guide-modifier">{{ m.name }}</small>
        {{ m.example|safe }}
    </div>
{% endfor %}

<div class="style-guide-html">
    <pre>{{ section.example|forceescape }}</pre>
</div>
</article>
```

Depending on the template you are using, apply some styles to it.

Note: This example uses [less](#) and can be installed via [bower](#):

```
bower install --save classy-style-guide
```

```
@import 'classy-style-guide/components.style-guide.less';
```

Contributing

Don't hesitate to create a [GitHub issue](#) for any **bug** or **suggestion**.

Changelog

3.1 0.4.0

- Add documentation.

3.2 0.3.0

- Add missed files into source distribution.
- Remove executable bit from py files.
- Refactor test suite; use pytest-flask for testing.

3.3 0.2.0

- Set package status to beta :tada:
- Add styleguide example into readme (#5).
- Fix python 2.6 and python 3 compatibility.
- Add compatible python versions into package metadata.

3.4 0.1.5

- Use *flask_styleguide* instead of *flask.ext.styleguide* in package imports since the *import flask.ext.foo* syntax is being deprecated for Flask 1.0, as per [mitsuhiko/flask#1135](#).

3.5 0.1.4

- Replace deprecated `codecs.open` with `io.open`.

3.6 0.1.3

- Parse package at runtime to get version string, as per #6.

3.7 0.1.2

- Ensure the html examples has no extra leading whitespaces by dedenting them during render.

3.8 0.1.1

- First release on PyPi.

3.9 0.1.0

- Proof of concept.