

---

# **jenkins\_badges Documentation**

***Release 1.2.0***

**Jeremy Arr**

**Jan 30, 2018**



---

## Contents

---

<b>1</b>	<b>Supported Badges</b>	<b>3</b>
<b>2</b>	<b>Get it now</b>	<b>5</b>
<b>3</b>	<b>Jenkins Requirements</b>	<b>7</b>
<b>4</b>	<b>Quickstart</b>	<b>9</b>
<b>5</b>	<b>Placing into a Readme File</b>	<b>11</b>
<b>6</b>	<b>WSGI Example</b>	<b>13</b>
<b>7</b>	<b>Comparison with Shields.io API</b>	<b>15</b>
<b>8</b>	<b>Responsiveness</b>	<b>17</b>
<b>9</b>	<b>Troubleshooting</b>	<b>19</b>
<b>10</b>	<b>API</b>	<b>21</b>
<b>11</b>	<b>Contributing</b>	<b>23</b>
<b>12</b>	<b>Project info</b>	<b>25</b>
	<b>Python Module Index</b>	<b>27</b>





# jenkins\_badges

Latest Version: v 1.2.0

`jenkins_badges` is a small flask app that serves dynamic badge images based on data from Jenkins CI.

- *Supported Badges*
- *Get it now*
- *Jenkins Requirements*
- *Quickstart*
- *Placing into a Readme File*
- *WSGI Example*
- *Comparison with Shields.io API*
- *Responsiveness*
- *Troubleshooting*
- *API*
- *Contributing*
- *Project info*



# CHAPTER 1

---

## Supported Badges

---

Endpoint	Examples	Default
coverage/<JenkinsJob>		80% +
		20%-80%
		< 20%
		error getting coverage data





## CHAPTER 2

---

Get it now

---

### 2.1 With pip:

```
$ pip install jenkins_badges
```



## CHAPTER 3

---

### Jenkins Requirements

---

*jenkins\_badges* communicates with your Jenkins instance over the [Jenkins API](#) . You need to either set up up the *anonymous* user in Jenkins with read access or supply *jenkins\_badges* with the credentials of a jenkins user who has read access.

For the coverage badge to work, your Jenkins instance must have the [Cobertura plugin](#) installed with coverage data being supplied to it after every successful build.



*jenkins\_badges* needs to be provided with information about your jenkins instance. This can be provided as arguments to the *create\_app* function or via a configuration file.

### 4.1 Supplying configuration parameters directly

1. create and run the app

```
import jenkins_badges

#path to your jenkins instance
base_url = "https://example.com/jenkins"

# not required if anonymous jenkins user has read access
username = "apiuser" #a user with read access
token = "6c3bde145bcda49402b6914f2353a734" #user's token

app = jenkins_badges.create_app(base_url=base_url,
                                username=username,
                                token=token)

app.run()
```

Output:

```
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

2. Your coverage badge image should be accessible at *http://127.0.0.1:5000/coverage/<JenkinsJobName>*

### 4.2 Supplying a configuration file

1. Create a configuration file

```
# /home/ubuntu/.jenkins_badges
JENKINS_BASE_URL = "https://example.com/jenkins"

# not required if anonymous jenkins user has read access
JENKINS_USERNAME = "apiuser" #a user with read access
JENKINS_TOKEN = "6c3bde145bcda49402b6914f2353a734" #user's token
```

2. Create a JENKINS\_BADGES\_SETTINGS environmental variable with the path to the configuration file:

Linux:

```
$ export JENKINS_BADGES_SETTINGS=/home/ubuntu/.jenkins_badges
```

3. create and run the app

```
import jenkins_badges

app = jenkins_badges.create_app(from_envvar=True)
app.run()
```

Output:

```
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

4. Your coverage badge image should be accessible at <http://127.0.0.1:5000/coverage/<JenkinsJobName>>

---

### Placing into a Readme File

---

Displaying a badge is as easy as placing a reference into your readme file.

If your readme file uses Markdown:

```
! [Coverage] (http://127.0.0.1:5000/coverage/<JenkinsJobName>)
```

If your readme file uses restructuredText:

```
.. image:: http://127.0.0.1:5000/coverage/<JenkinsJobName>
```





## CHAPTER 6

---

### WSGI Example

---

Just like any Flask app, a *jenkins\_badges* app can be placed on a server with WSGI support:

```
#exampleapp.wsgi

import os

#tell jenkins_badges where the config file is located
os.environ['JENKINS_BADGES_SETTINGS'] = '/home/ubuntu/.jenkins_badges'

from jenkins_badges import create_app

#name of app must be "application"
application = create_app()
```



---

### Comparison with Shields.io API

---

[shields.io](https://shields.io) has a simple API for accessing Jenkins coverage data, **providing your Jenkins anonymous user is granted read access**:

*<https://img.shields.io/jenkins/c/<scheme>/<jenkins host>/job/<Jenkins Job>.svg>*

*shields.io* also hard codes badge colours and the number of decimal points. *jenkins\_badges* is more suited if you want finer control of how your coverage badge is displayed or if you only allow authenticated users to access your jenkins instance.



## CHAPTER 8

---

### Responsiveness

---

*jenkins\_badges* serves badge images with a “maxAge” *cache-control* header value of 30 seconds. It does not perform redirects to *shields.io* due to GitHub’s well known [badge caching problems](#). As long as the server hosting your documentation respects *cache-control* headers, your badge should update on a page refresh after a jenkins build.

The responsiveness of images served by *jenkins\_badges* has been successfully tested on readme pages hosted by GitHub.



### 9.1 Coverage error badge displayed

Problem communicating with Jenkins.

You can test out whether *jenkins\_badges* will be able to communicate with Jenkins by performing the following API request:

Linux:

```
$ curl http<s>://<path to your jenkins instance>/job/<job name>/lastSuccessfulBuild/  
↪cobertura/api/json/?depth=2
```

Sample Output:

```
{ "_class": "hudson.plugins.cobertura.targets.CoverageResult", "results": { "children": [{  
↪ "children": [{ }], "elements": [{ }, { }, { }, { }], "name": "marbl" }], "elements": [{ "denominator  
↪ ": 1.0, "name": "Packages", "numerator": 1.0, "ratio": 100.0 }, { "denominator": 1.0, "name":  
↪ "Files", "numerator": 1.0, "ratio": 100.0 }, { "denominator": 1.0, "name": "Classes",  
↪ "numerator": 1.0, "ratio": 100.0 }, { "denominator": 5.0, "name": "Lines", "numerator": 4.0,  
↪ "ratio": 80.0 }, { "denominator": 0.0, "name": "Conditionals", "numerator": 0.0, "ratio": 100.  
↪ 0 } ], "name": "Cobertura Coverage Report" } }
```





The jenkins\_badges API is pretty simple, consisting of one function. Under the covers, the app is divided into flask blueprints, which allows it to be easily extended in the future. *jenkins\_badges* is a small flask app that serves dynamic badge images based on data from Jenkins CI.

`jenkins_badges.create_app` (*from\_envvar=False*, *base\_url=None*, *username=None*, *token=None*,  
*coverage\_yellow=80*, *coverage\_red=20*, *coverage\_decimal\_points=2*)  
creates the flask application object

### Parameters

- **from\_envvar** (*bool*) – if True, configuration parameters are sourced from file referenced by the local JENKINS\_BADGES\_SETTINGS environmental variable. if False, configuration parameters are sourced from the arguments.
- **base\_url** (*str*) – url of Jenkins instance. Must be supplied if from\_envvar=False.
- **username** (*str*) – username of Jenkins user
- **token** (*str*) – token of Jenkins user
- **coverage\_yellow** (*int*) – threshold for displaying yellow coverage colour. Coverage below this threshold and above the coverage\_red threshold will be shown with a yellow background. Coverage above this threshold will be shown with a green background.
- **coverage\_red** (*int*) – threshold for displaying red coverage colour. Coverage below this threshold will be shown with a red background.
- **coverage\_decimal\_points** (*int*) – number of decimal points displayed on badge for coverage figure.

**Returns** a flask application object



## CHAPTER 11

---

### Contributing

---

Source code for *jenkins\_badges* can be found [here](#) . New issues, feature requests and pull requests are all welcome.



## 12.1 Changelog

### 12.1.1 1.0.0 (2017-06-30)

- Initial release.

### 12.1.2 1.1.0 (2017-07-03)

- Added coverage threshold and decimal point configuration options

### 12.1.3 1.1.1 (2017-07-03)

- default arguments supplied in `__init__.py` instead of `default_settings.py`
- version string added

### 12.1.4 1.2.0 (2017-07-03)

- now compatible with python 2.7+

## 12.2 License

MIT License

Copyright (c) 2017 Jeremy Arr

Permission is hereby granted, free of charge, to any person obtaining a copy

of this software **and** associated documentation files (the "**Software**"), to deal **in** the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, **and/or** sell copies of the Software, **and** to permit persons to whom the Software **is** furnished to do so, subject to the following conditions:

The above copyright notice **and** this permission notice shall be included **in all** copies **or** substantial portions of the Software.

THE SOFTWARE IS PROVIDED "**AS IS**", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 12.3 Authors

### 12.3.1 Lead

- Jeremy Arr @jeremyarr

## 12.4 Kudos

- Idea came from mnpk's [jenkins-coverage-badge](#) written in nodeJS.
- [shields.io](#) for providing scalable badges over a clean API
- [Jenkins](#) for being...jenkins

### j

`jenkins_badges`, [21](#)





### C

`create_app()` (in module `jenkins_badges`), [21](#)

### J

`jenkins_badges` (module), [1](#), [21](#)