
libgithub Documentation

Release 1.0

Matt Sparks

October 05, 2014

1	Requirements and Installation	3
2	Usage	5
3	API	7
3.1	libgithub.Badge	7

libgithub is a JavaScript library for displaying GitHub data on websites. It uses the [GitHub API v2](#).

Currently libgithub only generates ‘badges’, which look something like this:

😊 [mspark/alphasign](#)



Matt Sparks committed **3d3b96e078** about 8 months ago

📦 *"Changing setup.py version to 1.0"*

(0 added, 0 removed, 1 modified) [Show files](#)

😊 [mspark/alphasign](#)



Matt Sparks committed **155b779d73** about 8 months ago

📦 *"Adding a much-needed complete example"*

(0 added, 0 removed, 3 modified) [Hide files](#)

Modified:

- alphasign/__init__.py
- alphasign/string.py
- docs/index.rst

😊 [mspark/alphasign](#)



Matt Sparks committed **82f40a10e1** about 8 months ago

📦 *"Documenting speeds"*

(0 added, 0 removed, 1 modified) [Show files](#)

libgithub is available from [GitHub](#) as a single JavaScript file. A ‘minified’ version of the library is available as well. CSS styles are also included with the distribution and can be customized.

Requirements and Installation

libgithub relies on several other JavaScript libraries to function properly, all of which are conveniently included in the libgithub distribution:

- jQuery
- humane.js
- date.js
- md5.js

These libraries must be loaded along with `libgithub.min.js` in the HTML files which use libgithub.

Thus, to load libgithub for use, simply include in your HTML file:

```
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" src="humane.js"></script>
<script type="text/javascript" src="date.js"></script>
<script type="text/javascript" src="md5.js"></script>
<script type="text/javascript" src="libgithub.min.js"></script>
```

You may need to modify the file paths as appropriate, depending on your installation. To get a style similar to the screenshot above, you will also need to include the CSS stylesheet, using something like:

```
<link type="text/css" rel="stylesheet" href="css/libgithub.css" />
```

Again, modify the path as appropriate. This stylesheet references images also included with the libgithub distribution that are expected to be in a directory called 'img' by default.

Usage

Here is a simple example for creating a badge:

```
<script type="text/javascript">
$(window).load(function () {
  var b = new libgithub.Badge('mspark', 'alphasign');
  b.targetIs('#commit');
});
</script>

<div id="commit"></div>
```

This example uses jQuery to create a badge when the window loads. The output is displayed in the element selected by '#commit', which is the `<div id="commit">`.

To display multiple commits or change other various settings, see the [libgithub.Badge](#) API.

3.1 libgithub.Badge

This class is used to create and display GitHub commit badges.

3.1.1 Reference

class `libgithub.Badge` (*username*, *repo* [, *commitId*])
Construct a new badge object.

Parameters

- **username** – GitHub username
- **repo** – Repository name
- **commitId** – ref name or commit ID (default: 'master')

gravatarSize ()

Get the [Gravatar](#) size. Defaults to 60 pixels.

gravatarSizeIs (*size*)

Set the [Gravatar](#) size.

Parameters *size* – width of Gravatar in pixels

startHidden ()

Current `startHidden` setting. If true, the file list is hidden when the badge is created. Defaults to true.

startHiddenIs (*setting*)

Set the `startHidden` setting.

Parameters *setting* – new setting. True = hide file list, False = show file list

target ()

Get the current target.

targetIs (*selector*)

Set the target HTML element via a jQuery selector.

This issues a request to GitHub for commit data and creates a callback to display the badge.

The badge is appended to the inside of the element selected by the selector.

Parameters *selector* – string selector to find the target HTML element. See the [jQuery documentation](#) for more information about valid selectors.

3.1.2 Examples

Here is a simple example for creating a badge:

```
<script type="text/javascript">
$(window).load(function () {
  var b = new libgithub.Badge('mspark', 'alphasign');
  b.targetIs('#commit');
});
</script>

<div id="commit"></div>
```

To show the file list by default, simply toggle the `startHidden` setting like so:

```
var b = new libgithub.Badge('mspark', 'alphasign');
b.startHiddenIs(false);
b.targetIs('#commit');
```

The `commitId` parameter in the badge constructor can be a ref of any kind: a specific commit ID, partial or complete; a branch name; or a tag name. Creating a badge for a specific commit is easy:

```
var b = new libgithub.Badge('mspark', 'alphasign', 'c5f6c7a2');
b.targetIs('#commit');
```

G

`gravatarSize()` (libgithub.Badge method), [7](#)
`gravatarSizeIs()` (libgithub.Badge method), [7](#)

L

`libgithub.Badge` (built-in class), [7](#)

S

`startHidden()` (libgithub.Badge method), [7](#)
`startHiddenIs()` (libgithub.Badge method), [7](#)

T

`target()` (libgithub.Badge method), [7](#)
`targetIs()` (libgithub.Badge method), [7](#)