Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

visionmedia / **express-messages**

Express flash notification message rendering

40 commits | 1 branch | 0 packages | 2 releases | 6 contributors

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

Ryan bump version for doc change — Latest commit d1db1cb on Apr 25, 2016

test	Revert flattening of messages object	5 years ago
.gitignore	Fixed tests	9 years ago
.npmignore	Updated .npmignore	5 years ago
History.md	Release 0.0.2	9 years ago
Readme.md	clarifying usage instructions for custom templates	4 years ago
index.js	Revert flattening of messages object	5 years ago
package.json	bump version for doc change	4 years ago

Readme.md

Express Messages

The *express-messages* module provides flash notification rendering.

Installation

```
$ npm install express-messages
```

Usage

Express 2.x

To use simply assign it to a dynamic helper:

```
app.dynamicHelpers({ messages: require('express-messages') });
```

Express 3+

Install connect-flash and add them as middleware:

```
app.use(require('connect-flash')());
app.use(function (req, res, next) {
  res.locals.messages = require('express-messages')(req, res);
  next();
});
```

Adding Messages

On the server:

```
req.flash("info", "Email queued");
req.flash("info", "Email sent");
req.flash("error", "Email delivery failed");
```

For further information see connect-flash.

Rendering Messages

Call the `messages()` function as specified by your rendering engine:

EJS:

```
<%- messages() %>
```

Jade:

```
!= messages()
```

Which will output the HTML:

```
<div id="messages">
  <ul class="info">
    <li>Email queued</li>
    <li>Email sent</li>
  </ul>
  <ul class="error">
    <li>Email delivery failed</li>
  </ul>
</div>
```

Using a custom template

Alternatively you can specify a custom template (a file in the views directory of your Express app).

Add a message template

For example, lets use the below custom message template named `my_message_template` .

EJS (`my_message_template.ejs`):

```
<div id="messages">
<% Object.keys(messages).forEach(function (type) { %>
  <ul class="<%= type %>">
  <% messages[type].forEach(function (message) { %>
    <li><%= message %></li>
  <% }) %>
  </ul>
<% }) %>
</div>
```

Jade (`my_message_template.jade`):

```
.messages
  each type in Object.keys(messages)
    ul(class="#{type}")
      each message in messages[type]
        li= message
```

Call the message template

Next, pass the template name, `my_message_template` , as a parameter to the `messages()` function.

EJS:

```
<%- messages('my_message_template', locals) %>
```

Jade:

```
!= messages('my_message_template', locals)
```

The message template will receive an object called `messages` of the form:

```
{
  "info" : [
    "Email queued",
    "Email sent"
  ],
  "error": [
    "Email delivery failed"
  ]
}
```

Running Tests

```
$ npm test
```

License

(The MIT License)

Copyright (c) 2010 TJ Holowaychuk <tj@vision-media.ca>

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.