express-ejs-layouts

2.5.0 • Public • Published 2 years ago

Readme

Explore BETA

0 Dependencies

87 Dependents

13 Versions

Install

```
> npm i express-ejs-layouts
```

Weekly Downloads

9,228

Version	License
2.5.0	none
Unpacked Size	**Total Files**
11.4 kB	8

Issues
3

Pull Requests
1

Homepage
github.com/Soarez/express-ejs-layouts#readme

Repository
github.com/Soarez/express-ejs-layouts

Last publish
2 years ago

Collaborators


Try on RunKit

Report a vulnerability

express-ejs-layouts

> Layout support for ejs in express

Installation

```
$ npm install express-ejs-layouts
```

Example

Check the example folder.

```
1 git clone git@github.com:Soarez/express-ejs-layouts.git
```

1. git clone git@github.com:soarez/express-ejs-layouts.git
2. cd express-ejs-layouts
3. npm install
4. node example
5. Open **http://localhost:3000/**

Usage

```
var express = require('express');
var expressLayouts = require('express-ejs-layouts');

var app = express();

app.set('view engine', 'ejs');

app.use(expressLayouts);

app.get('/', function(req, res) {
  var locals = {
    title: 'Page Title',
    description: 'Page Description',
    header: 'Page Header'
  };
  res.render('the-view', locals);
});

app.listen(3000);
```

contentFor

A view

```
tyler

<%- contentFor('foo') %>
club
<%- contentFor('bar') %>
fight
```

With a layout

```
<%-bar%> <%-foo%>
<%-body%>
```

Renders

```
fight club
tyler
```

As another example, consider this view:

```
foo
<%- contentFor('pageSectionA') %>
bar
<%- contentFor('pageSectionB') %>
baz
```

Using it with this layout:

```
<div class="header"><%- pageSectionA %></div>
<div class="body"><%- body %></div>
<div class="footer"><%-defineContent('pageSectionB')%></div>
```

Will render:

```
<div class="header">bar</div>
<div class="body">foo</div>
<div class="footer">baz</div>
```

Notice that the difference between using `<%- pageSectionA %>` and `<%- defineContent('pageSectionA')%>` is that the former will generate an error if the view doesn't define content for this section.

Script blocks extraction

Script blocks extraction

If you like to place all the script blocks at the end, you can do it like this:

```
app.set("layout extractScripts", true)
```

A view

```
something<script>somejs<script>something
```

With a layout

```
<body>
  <%- body %>
  <%- script %>
</body>
```

Renders

```
<body>
  somethingsomething
  <script>somejs<script>
</body>
```

Enabling invididually:

```
req.render('view', { extractScripts: true })
```

When the `"layout extractScripts"` option is activated, scripts defined in views will be extracted (won't be a part of `body`) and will be available for use in the layout through the variable `scripts`.

Another example:

This view:

```
<script src="/b.js" />
<div>foo</div>
```

```
<script src="/a.js" />
<div>bar</div>
<script src="/c.js" />
```

Used with this layout:

```
<div class="main">
<%- body %>
</div>
<!-- place the scripts at the end of the html page -->
<%- script %>
```

Will render:

```
<div class="main">
<div>foo</div>
<div>bar</div>
</div>
<!-- place the scripts at the end of the html page -->
<script src="/b.js" />
<script src="/a.js" />
<script src="/c.js" />
```

Style blocks extraction

Works exactly like script blocks extraction except:

- Supported tags are `<link rel="stylesheet" …>` and `<style …>`
- The option is named `extractStyles`
- The template variable in layout is `style`

Meta blocks extraction

Works exactly like script blocks extraction except:

- Supported tags are `<meta …>` and `<meta …/>`
- The option is named `extractMetas`
- The template variable in layout is `meta`

Set custom default layout

By default 'layout.ejs' is used. If you want to specify your custom layout (e.g. 'layouts/layout.ejs'), just set `layout` property in express app settings.

```
app.set('layout', 'layouts/layout');
```

Set custom layout for single render

Just pass `layout` as render locals object.

```
app.get('/', function(req, res) {
  res.render('the-view', { layout: 'specific-layout' });
);
```

Optional sections

In a layout, you can have optional sections using `defineContent` : Unspecified section content defaults to `''` .

```
1
<%-defineContent('a')%>
2
<%-defineContent('b')%>
3
```

with a view:

```
<%- contentFor('a') %>
1.5
```

will render:

```
1
1.5
2
3
```

Running tests

Clone the repo and run:

```
$ npm test
```

License

MIT

Keywords

express layout ejs

Support

Help

Community

Advisories

Status

Contact npm

Company

About

Blog

Press

Terms & Policies

Policies

Terms of Use

Code of Conduct

Privacy