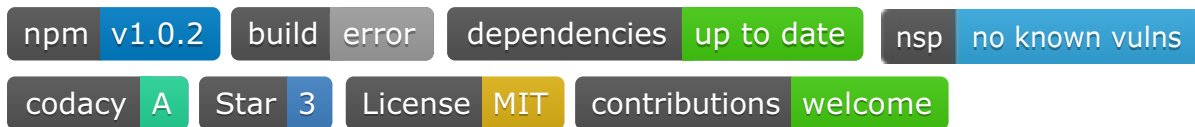


Helium.js

Making your React application lighter!



Helium.js is a node package that helps make your React application isomorphic and optimized.

- **Currently:** Includes server side rendering with support for React Router v4 and Redux v3
- **Coming Soon:** Optimization for webpack bundles

Prerequisites

You will need to have react/react-dom and the babel-cli installed as dependencies.

```
$ npm install react react-dom babel-cli --save
```

Installation

```
$ npm install helium.js --save
```

Usage

Hydrating on Client Side

```
/* Replace render with helium method
inside the index file of React application */

import helium from 'helium.js/react';

helium(
  <BrowserRouter>
    <App/>
  </BrowserRouter>,
  'root'
);
```

(with Redux)

```
/* Replace render with helium method
inside the index file of React application */

import helium, { createStore } from 'helium.js/react';

helium(
  <Provider store={createStore(reducer)}>
    <BrowserRouter>
      <App/>
    </BrowserRouter>
  </Provider>,
  'root'
);
```

(with Redux and Middlewares)

```
/* Replace render with helium method
inside the index file of React application.
Declare your middlewares as usual and pass
```

```
in as a second parameter to createStore invocation */

import helium, { createStore } from 'helium.js/react';

// declare your middlewares

helium(
  <Provider store={createStore(reducer, middleware)}>
    <BrowserRouter>
      <App/>
    </BrowserRouter>
  </Provider>,
  'root'
);
```

Rendering on Server Side

Option 1: Automation with CLI

Have your server file automatically generated by answering questions using our cli.

To start up the cli, do one of the following:

* Type this command directly into your terminal

```
$ ./node_modules/.bin/he
```

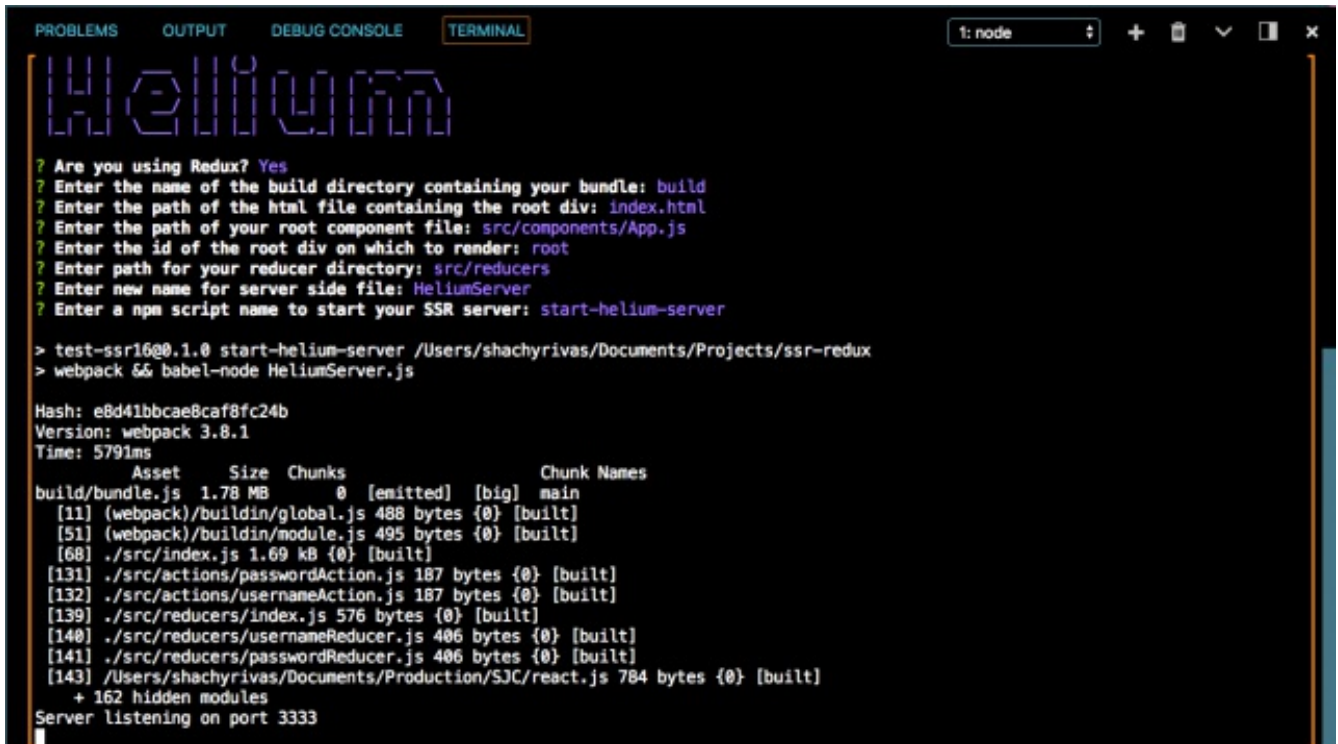
* Add a script to your package.json and run the script

```
"scripts": {
  "start-helium-server": "./node_modules/.bin/he",
},
```

```
$ npm run he
```

* Install globally

```
$ npm i -g helium.js  
$ he
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: node  
Helium  
? Are you using Redux? Yes  
? Enter the name of the build directory containing your bundle: build  
? Enter the path of the html file containing the root div: index.html  
? Enter the path of your root component file: src/components/App.js  
? Enter the id of the root div on which to render: root  
? Enter path for your reducer directory: src/reducers  
? Enter new name for server side file: HeliumServer  
? Enter a npm script name to start your SSR server: start-helium-server  
  
> test-ssr16@0.1.0 start-helium-server /Users/shachyirivas/Documents/Projects/ssr-redux  
> webpack && babel-node HeliumServer.js  
  
Hash: e8d41bbcae8caf8fc24b  
Version: webpack 3.8.1  
Time: 5791ms  
Asset Size Chunks Chunk Names  
build/bundle.js 1.78 MB 0 [emitted] [big] main  
[11] (webpack)/buildin/global.js 488 bytes {0} [built]  
[51] (webpack)/buildin/module.js 495 bytes {0} [built]  
[68] ./src/index.js 1.69 kB {0} [built]  
[131] ./src/actions/passwordAction.js 187 bytes {0} [built]  
[132] ./src/actions/usernameAction.js 187 bytes {0} [built]  
[139] ./src/reducers/index.js 576 bytes {0} [built]  
[140] ./src/reducers/usernameReducer.js 406 bytes {0} [built]  
[141] ./src/reducers/passwordReducer.js 406 bytes {0} [built]  
[143] /Users/shachyirivas/Documents/Production/SJC/react.js 784 bytes {0} [built]  
+ 162 hidden modules  
Server listening on port 3333
```

Option 2: Do it Yourself

```
/* Include this in your server file  
(the file in which you initialize your  
express application) */  
  
// import your root component
```

```

import App from './src/components/App.js';
const helium = require('helium.js');

// initialize your express application here

helium.init({
  // indicate the path to your main html file
  html: 'index.html',
  // specify the id to which your React application
  will be mounted on
  id: 'root',
  App,
});

// input api routes here

app.get('*', helium.serve);

```

(with Redux)

```

/* Include this in your server file
(the file in which you initialize your
express application) */

// import your root component and your reducer
import App from './src/components/App.js';
import reducer from './src/reducers';
const helium = require('helium.js');

// initialize your express application here

helium.init({
  // indicate the path to your main html file
  html: 'index.html',
  // specify the id to which your React application

```

```
will be mounted on
  id: 'root',
  App,
  reducer,
});

// input api routes here

app.get('*', helium.serveRedux);
```

Running Your Application

If cli was not used, be sure to add a script to your package.json.

```
"scripts": {
  "start-helium-server":
    "./node_modules/.bin/webpack && babel-node
    HeliumServer.js",
},
```

```
# To run your application, type the following into
your terminal
$ npm run start-helium-server
```

Contributing

If you would like to contribute, submit a [pull request](#) and update the README.md with details of changes.

Versioning

We use [SemVer](#) for versioning. For the versions available, see the [tags on](#)

this repository.

Authors

- **Shachy Rivas** – [shachyjr](#)
- **Chris Li** – [cli53](#)
- **Julie Moon** – [juliemoon](#)

License

This project is licensed under the MIT License – see the [LICENSE.md](#) file for details