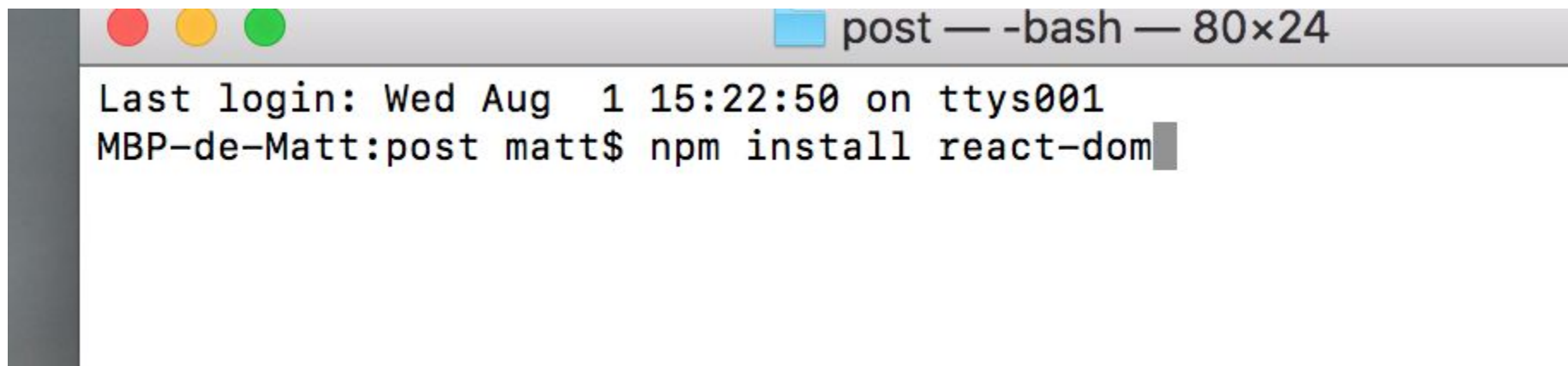
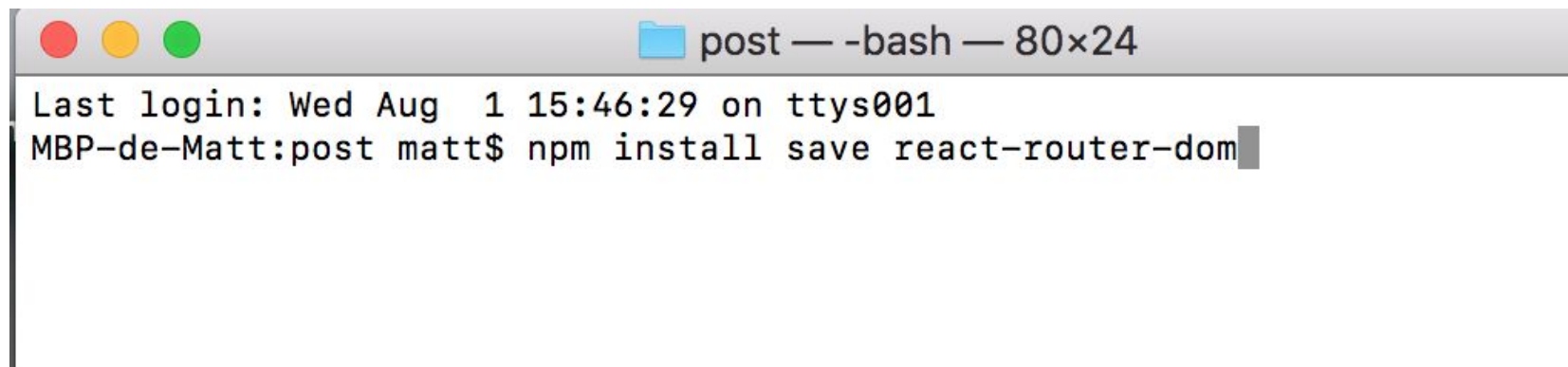


installation 1

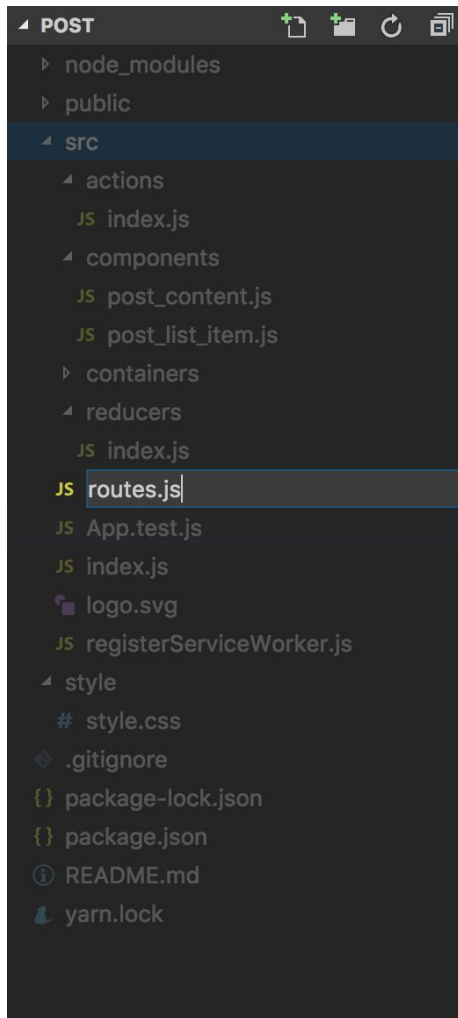
A terminal window with a title bar containing three colored circles (red, yellow, green) and a folder icon followed by the text "post — -bash — 80x24". The terminal text shows a login message and a command to install react-dom.

```
Last login: Wed Aug  1 15:22:50 on ttys001
MBP-de-Matt:post matt$ npm install react-dom
```

installation 2

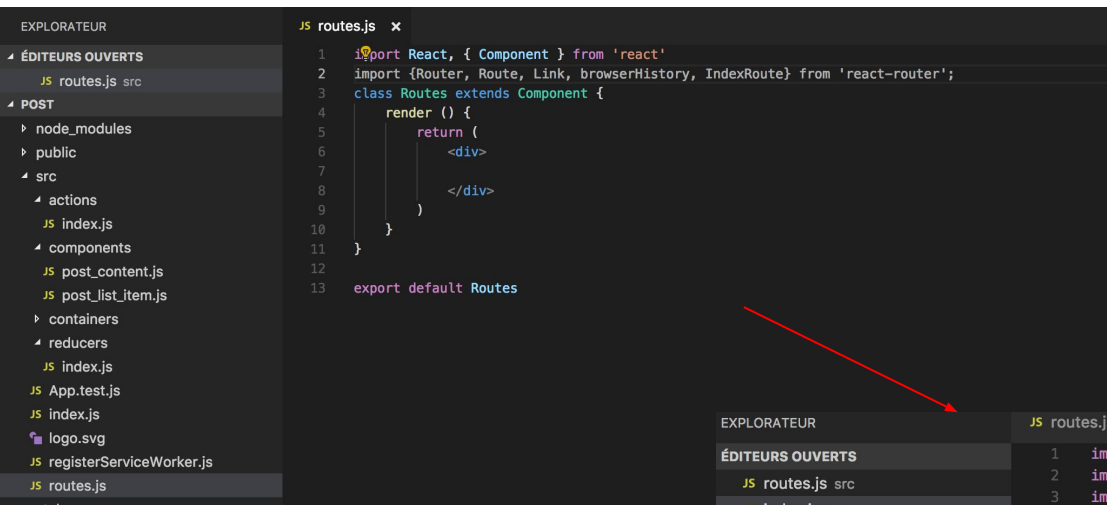
A terminal window with a title bar containing three colored circles (red, yellow, green) and a folder icon followed by the text "post — -bash — 80x24". The terminal text shows a login message and a command to install save react-router-dom.

```
Last login: Wed Aug  1 15:46:29 on ttys001
MBP-de-Matt:post matt$ npm install save react-router-dom
```



On créer un nouveau point d'entrée en créant un fichier route.js

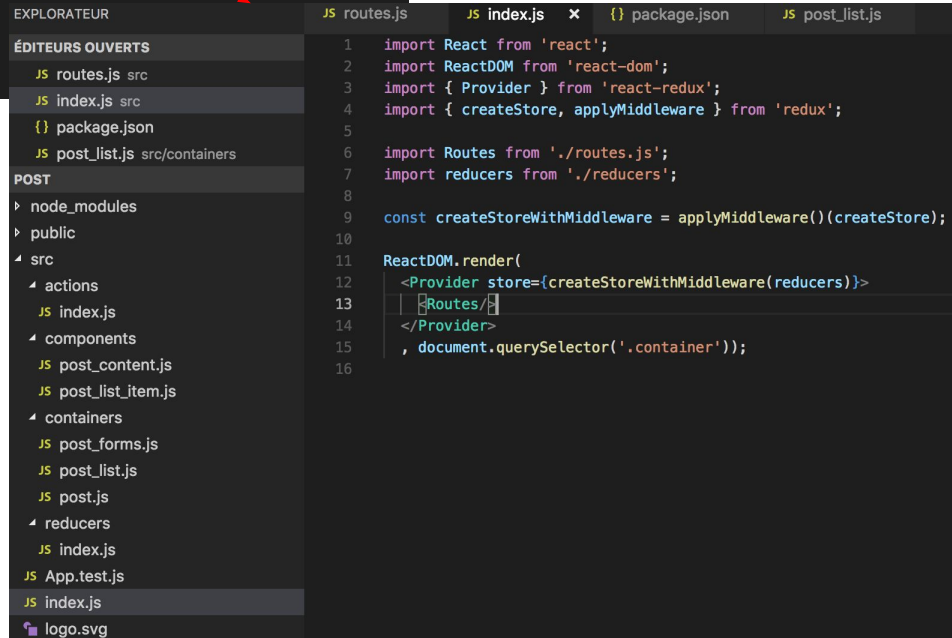
Create a new entry point by creating a route.js file



```
JS routes.js x
1 import React, { Component } from 'react'
2 import {Router, Route, Link, browserHistory, IndexRoute} from 'react-router';
3 class Routes extends Component {
4   render () {
5     return (
6       <div>
7
8       </div>
9     )
10   }
11 }
12
13 export default Routes
```

“react-router” va être comme notre nouveau point d’entrée de notre application, il va selon l’url qu’on lui donne importer et afficher le composant voulus.

React-router" is going to be like our new entry point of our application, it goes according to the url that one gives him to import and to display the desired component.



```
EXPLORATEUR JS routes.js JS index.js x {} package.json JS post_list.js
ÉDITEURS OUVERTS
JS routes.js src
JS index.js src
{} package.json
JS post_list.js src/containers
POST
node_modules
public
src
actions
JS index.js
components
JS post_content.js
JS post_list_item.js
containers
JS post_forms.js
JS post_list.js
JS post.js
reducers
JS index.js
JS App.test.js
JS index.js
logo.svg
registerServiceWorker.js
JS routes.js

1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import { Provider } from 'react-redux';
4 import { createStore, applyMiddleware } from 'redux';
5
6 import Routes from './routes.js';
7 import reducers from './reducers';
8
9 const createStoreWithMiddleware = applyMiddleware()(createStore);
10
11 ReactDOM.render(
12   <Provider store={createStoreWithMiddleware(reducers)}>
13     <Routes/
14   </Provider>
15   , document.querySelector('.container'));
16
```

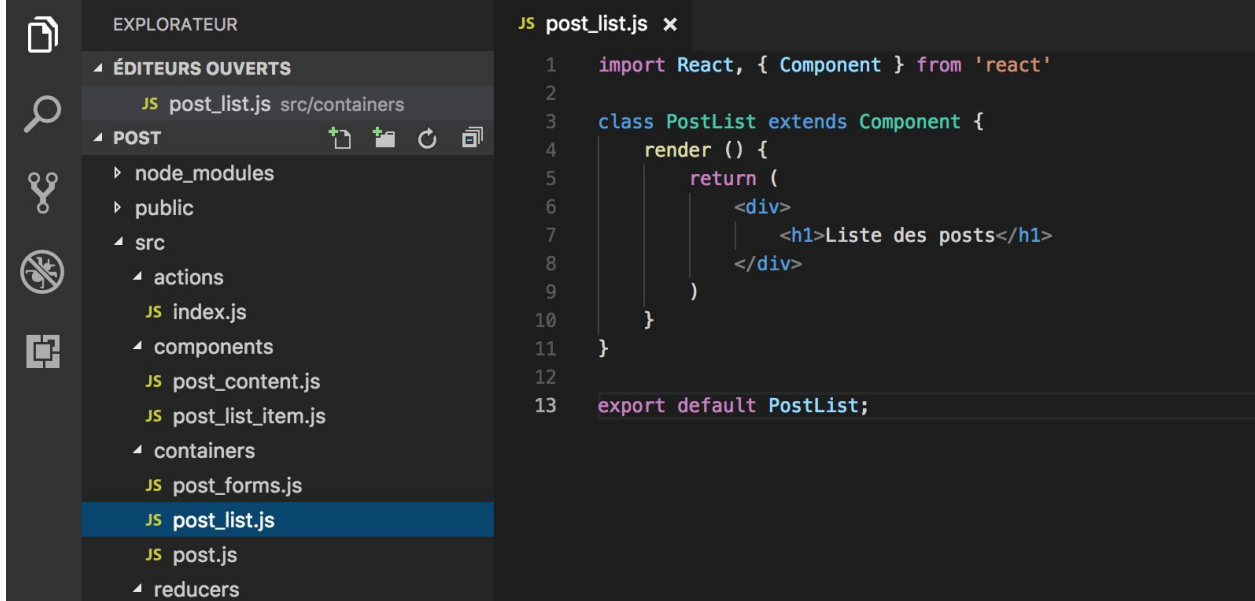
implementation

The screenshot displays a code editor with a dark theme. On the left, the 'EXPLORATEUR' (Explorer) panel shows the project structure. Under 'ÉDITEURS OUVERTS' (Open Editors), the following files are listed: `JS routes.js src`, `JS index.js src`, `{ } package.json`, and `JS post_list.js src/containers`. Under the 'POST' section, a tree view shows the following structure:

- `node_modules`
- `public`
- `src`
 - `actions`
 - `index.js`
 - `components`
 - `post_content.js`
 - `post_list_item.js`
 - `containers`
 - `post_forms.js`
 - `post_list.js`
 - `post.js`
 - `reducers`
 - `index.js`
 - `App.test.js`
 - `index.js`
 - `logo.svg`
 - `registerServiceWorker.js`
 - `routes.js` (selected)
- `style`

The main editor area shows the code for `JS routes.js`. The code is as follows:

```
1 import React, { Component } from 'react'
2 import { BrowserRouter, Route } from 'react-router-dom'
3 import PostList from './containers/post_list';
4
5
6 class Routes extends Component {
7   render () {
8     return (
9       <BrowserRouter>
10         <Route path="/" component={PostList}/>
11       </BrowserRouter>
12     )
13   }
14 }
15
16
17 export default Routes
```



It's work :)

