



Gaurav Singhal

Load NPM Modules with React

Gaurav Singhal

Sep 12, 2020 • 4 Min read • 328 Views

Sep 12, 2020 • 4 Min read • 328 Views

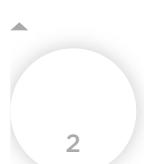
Web Development

Front End Web Dev...

Client-side Framew...

React

Introduction



- [Introduction](#)
- [Load the NPM Module Using the CLI](#)
- [Load the NPM Module Using the GitHub Repository](#)
- [Load the NPM Module Using the package.json File](#)

Introduction

NPM is short for *node package manager*, an online directory that contains the various already registered open-source packages. NPM modules consume the various functions as a third-party package when installed into an app using the NPM command `npm install`.

Third-party NPM modules are the unit of logic for a specific functionality or a full-fledged library, such as *reactstrap*, *material-ui*, *redux-form*, and so on. In this guide, you will learn how to load NPM

- [Conclusion](#)
- [Top ^](#)

modules in your app using the NPM command or the GitHub source code.

Load the NPM Module Using the CLI

Any NPM modules can be easily consumed in your app using the CLI command followed by the module name. Below is the syntax used to install the NPM module.

shell

```
1     npm install <module_name>
```

Try installing any module using the above CLI command. Let's install *reactstrap* for an example.

shell

```
1     npm install reactstrap
```

When you execute the above command, the related module folder will get added to the directory `node_modules` in your current app, and you will be able to use the components or functionality of the module anywhere in your app.

The installed package is [reactstrap](#), but you may wonder how to use it in the component. Import the module as given below.

```
jsx
1 import React, { Component } from "react";
2 // Import the module
3 import { Button, Popover, PopoverHeader, PopoverBody } from "reactstrap";
4
5 class SimplePopover extends Component {
6   render() {
7     const { popoverOpen } = this.state;
8
9     return (
10       <div>
11         // Using the module's component
12         <Popover
13           placement="bottom"
14           isOpen={popoverOpen}
15           target="mypopover"
16           toggle={this.togglePopover}
17         >
18           <PopoverHeader>This is popover title</PopoverHeader>
19           <PopoverBody>
20             This is simple popover content
21           </PopoverBody>
22         </Popover>
23       </div>
24     );
25   }
26 }
27 export default SimplePopover;
```

You will notice in the above code that there is one statement that imports the components from the installed module.

```
1 import { Button, Popover, PopoverHeader, PopoverBody } from "reactstrap";
```

You will be able to use it into the components after importing the required component from the module.

Load the NPM Module Using the GitHub Repository

You can use any NPM module by installing it using the CLI command, but alternatively, it can also be used directly by providing the GitHub repository URL as shown below.

```
1 npm install <git_repo_url>
```

Along with the CLI command, add the required GitHub repository URL of the module that you want to install.

```
1 npm install https://github.com/reactstrap/reactstrap.git
```

A folder gets created inside the directory `node_modules` by providing the GitHub URL along with the install command. And the package gets mentioned in `package.json`.

Load the NPM Module Using the `package.json` File

The file `package.json` contains all the installed dependencies' details, such as module and installed version. Still, if you want to install the module directly, you can do it by mentioning the two parameters, as shown below.

```
1      {  
2        "dependencies": {  
3          "package/module name": "version"  
4        }  
5      }
```

json

Specify the module you want to install and its required version so once the command `npm install` is executed, the mentioned package will be installed with all other packages.

So, in that case, `npm install package_name` does not require installing the package directly. This approach is quite handy when you know all the needed packages and versions.

Conclusion

Most JavaScript-based libraries or frameworks require the NPM packages to go ahead with app development. Hence the packages should be installed using one of the ways explained in this guide.