

# React JS

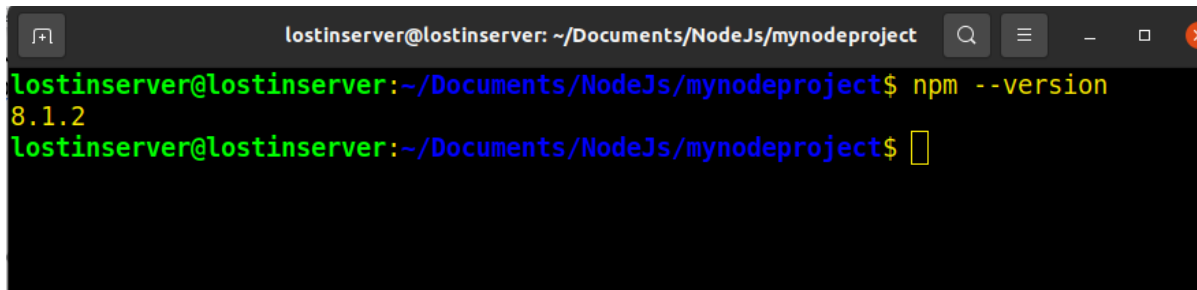
**React JS** is a free and open-source front-end JavaScript library for building user interfaces or UI components. It is developed by Facebook (Meta) in 2011.

## Installing react js in our system

Firstly we need to install **npm** package manager in our system so that we can move ahead.

*“ **npm** is a package manager for the JavaScript programming language maintained by npm.”*

```
$ sudo apt install npm    // installing npm
$ npm --version           // checking npm version
```

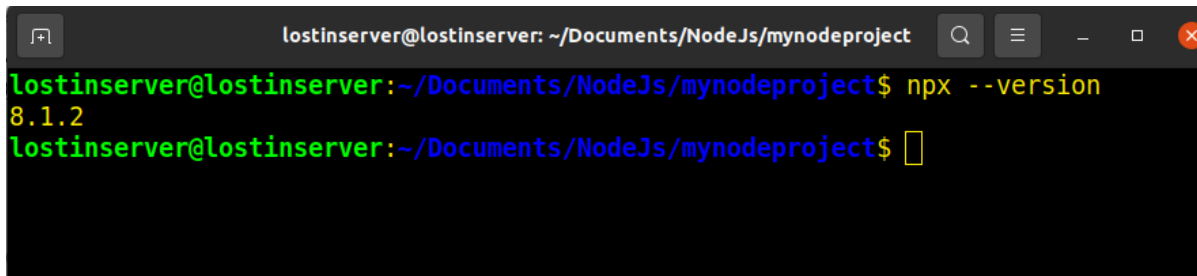
A terminal window with a dark background. The title bar shows 'lostinserver@lostinserver: ~/Documents/NodeJs/mynodeproject'. The prompt is 'lostinserver@lostinserver:~/Documents/NodeJs/mynodeproject\$'. The command 'npm --version' has been entered, and the output '8.1.2' is displayed on the next line. The prompt is now 'lostinserver@lostinserver:~/Documents/NodeJs/mynodeproject\$' with a cursor.

**npm** has the **npm** package installed by default. If it is not install using,

```
$ npm install -g npx      // installing npx package
$ npx --version           // checking npx version
```

**npx** is a package runner tool that comes with npm 5.2+ and helps to create a react app. We use this as through this we always can create a react app

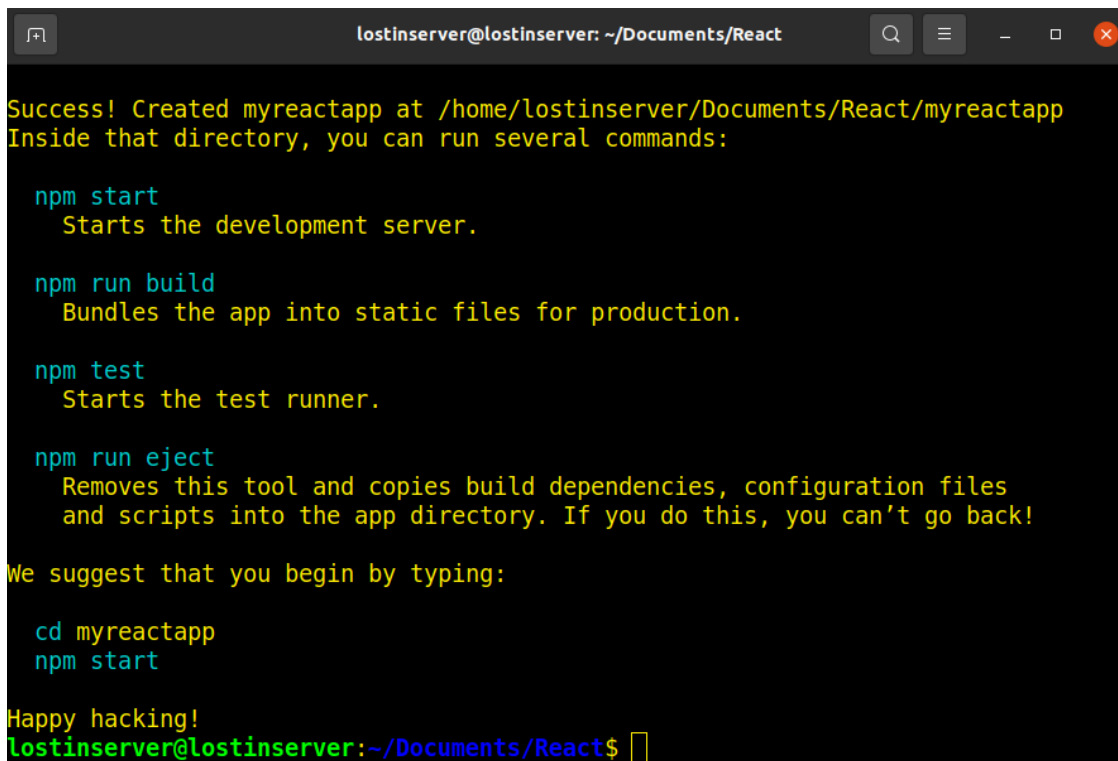
with updated packages and also the npx package is a lot smaller than the **create-react-app** utility.

A terminal window with a dark background. The title bar reads 'lostinserver@lostinserver: ~/Documents/NodeJs/mynodeproject'. The prompt is 'lostinserver@lostinserver:~/Documents/NodeJs/mynodeproject\$'. The command 'npx --version' has been entered, and the output '8.1.2' is displayed on the next line. The prompt is now 'lostinserver@lostinserver:~/Documents/NodeJs/mynodeproject\$' with a cursor.

```
lostinserver@lostinserver: ~/Documents/NodeJs/mynodeproject
lostinserver@lostinserver:~/Documents/NodeJs/mynodeproject$ npx --version
8.1.2
lostinserver@lostinserver:~/Documents/NodeJs/mynodeproject$
```

Now, we can create a new react app using,

**# npx create-react-app myreactapp**

A terminal window with a dark background. The title bar reads 'lostinserver@lostinserver: ~/Documents/React'. The output shows the successful creation of 'myreactapp' at '/home/lostinserver/Documents/React/myreactapp'. It lists several commands: 'npm start' (Starts the development server.), 'npm run build' (Bundles the app into static files for production.), 'npm test' (Starts the test runner.), and 'npm run eject' (Removes this tool and copies build dependencies, configuration files and scripts into the app directory. If you do this, you can't go back!). It suggests beginning by typing 'cd myreactapp' and 'npm start'. It ends with 'Happy hacking!' and the prompt 'lostinserver@lostinserver:~/Documents/React\$' with a cursor.

```
lostinserver@lostinserver: ~/Documents/React
Success! Created myreactapp at /home/lostinserver/Documents/React/myreactapp
Inside that directory, you can run several commands:

  npm start
    Starts the development server.

  npm run build
    Bundles the app into static files for production.

  npm test
    Starts the test runner.

  npm run eject
    Removes this tool and copies build dependencies, configuration files
    and scripts into the app directory. If you do this, you can't go back!

We suggest that you begin by typing:

  cd myreactapp
  npm start

Happy hacking!
lostinserver@lostinserver:~/Documents/React$
```

As we can see our react app has been successfully created.

Now, in order to print “**Hello React.js**” message we simply edit our **App.js** file located in our myreactapp/src as below,

```
JS App.js M X
src > JS App.js > App
1  import './App.css';
2
3  function App() {
4    return (
5      <div className="App">
6        Hello React.js
7      </div>
8    );
9  }
10
11 export default App;
12
```

(Optional) Let's also modify some css to make it a bit visible. So, let us modify App.css as,

```
JS App.js M # App.css M X
src > # App.css > ...
1  .App {
2    text-align: center;
3    line-height: 30vh;
4    font-weight: bold;
5    font-size: 40px;
6  }
7
```

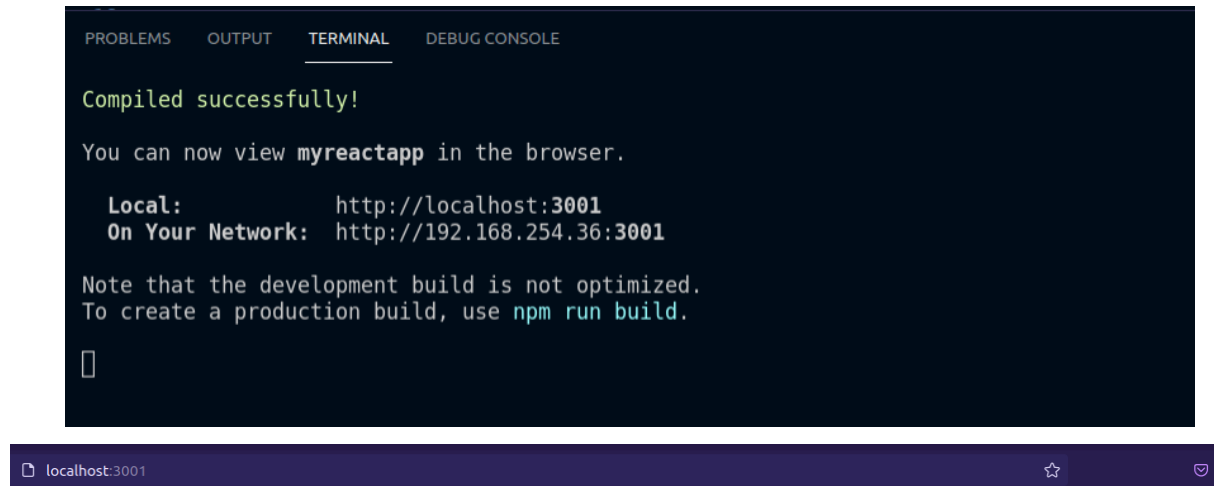
Default port for React is **3000**. We now change it to **3001** by simply using,

**\$ export PORT=3001**

*\* This should be done inside the project's directory i.e. myreactapp (here)*

After this, we can run npm to view our site,

**\$ npm start**



The image shows a screenshot of a Visual Studio Code terminal window. The terminal has tabs for 'PROBLEMS', 'OUTPUT', 'TERMINAL' (which is active), and 'DEBUG CONSOLE'. The output text in the terminal is as follows:

```
Compiled successfully!  
  
You can now view myreactapp in the browser.  
  
Local:      http://localhost:3001  
On Your Network:  http://192.168.254.36:3001  
  
Note that the development build is not optimized.  
To create a production build, use npm run build.  
  
□
```

Below the terminal window, there is a browser address bar showing 'localhost:3001' with a star icon and a shield icon on the right.

**Hello React.js**