React Js Documentation

**1 step :** install node js using below commands

Download the latest node js from google and install in the the system

For checking that node js has been install in your system execute the command.

>node –v

v10.15.1

## Installing ReactJS using webpack and babel

**2 step :** now you are able to create react app onto the node js so create you first app using command line as instructions given below.

D:\nodejs\_learning>mkdir rectApp

D:\nodejs\_learning>cd rectApp

To create any module, it is required to generate the **package.json** file. Therefore, after Creating the folder, we need to create a **package.json** file

To do so you need to run the **npm init** command from the command prompt.

**Note :** This command asks information about the module such as packagename, description, author etc. you can skip these using the –y option.

D:\nodejs\_learning\rectApp>npm init –y

Wrote to D:\nodejs\_learning\rectApp\package.json:

{

"name": "rectApp",

"version": "1.0.0",

"description": "",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"

},

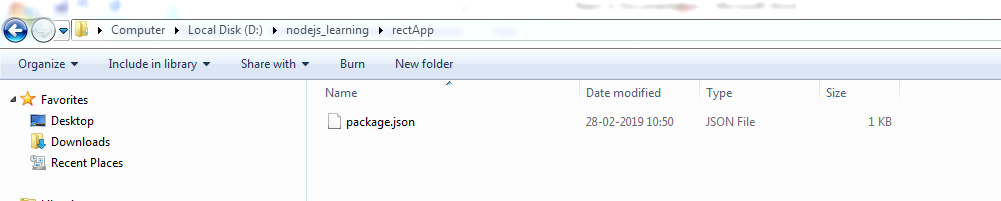
"keywords": [],

"author": "",

"license": "ISC"

}

Check your app folder --> (D:\nodejs\_learning\rectApp) you will get the package.json created into it.



Step 3 : install React and react dom

using **install react** and **react-dom** commands of npm respectively. You can add the packages we install, to **package.json** file using the **--save** option.

D:\nodejs\_learning\rectApp>npm install react –save

npm notice created a lockfile as package-lock.json. You should commit this file.

npm WARN rectApp@1.0.0 No description

npm WARN rectApp@1.0.0 No repository field.

+ react@16.8.3

added 7 packages from 3 contributors and audited 13 packages in 2.63s

found 0 vulnerabilities

D:\nodejs\_learning\rectApp>npm install react-dom –save

npm WARN rectApp@1.0.0 No description

npm WARN rectApp@1.0.0 No repository field.

+ react-dom@16.8.3

added 1 package and audited 26 packages in 1.877s

found 0 vulnerabilities

### Step 3 : Install webpack

### Since we are using webpack to generate bundler install webpack, webpack-dev-server and webpack-cli.

### D:\nodejs\_learning\rectApp>npm install webpack webpack-dev-server webpack-cli --save

### npm WARN rectApp@1.0.0 No description

### npm WARN rectApp@1.0.0 No repository field.

### npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.7 (node\_modules\fsevents):

### npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.7: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})

### + webpack-cli@3.2.3

### + webpack-dev-server@3.2.1

### + webpack@4.29.5

### added 527 packages from 305 contributors and audited 8699 packages in 55.638s

### found 0 vulnerabilities

### Step 4 - Install babel

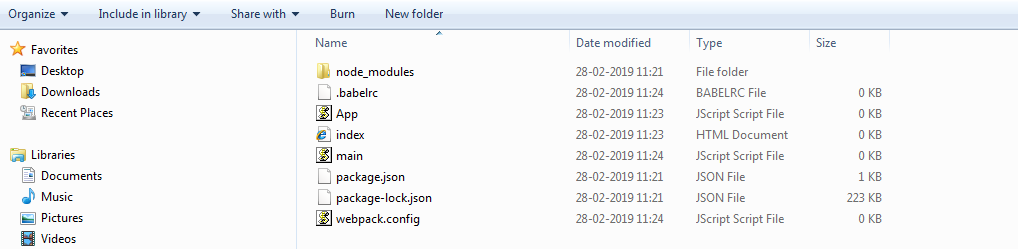
Install babel, and its plugins babel-core, babel-loader, babel-preset-env, babel-preset-react and, html-webpack-plugin

D:\nodejs\_learning\rectApp>npm install babel-core babel-loader babel-preset-env

babel-preset-react html-webpack-plugin --save-dev

### Step 5 - Create the Files

To complete the installation, we need to create certain files namely, index.html, App.js, main.js, webpack.config.js and, .***babelrc***. You can create these files manually or, using **command prompt**.



### Step 6 - Set Compiler, Server and Loaders

Open **webpack-config.js** file and add the following code. We are setting webpack entry point to be main.js. Output path is the place where bundled app will be served. We are also setting the development server to **8001** port. You can choose any port you want.

const path = require('path');

const HtmlWebpackPlugin = require('html-webpack-plugin');

module.exports = {

entry: './main.js',

output: {

path: path.join(\_\_dirname, '/bundle'),

filename: 'index\_bundle.js'

}, devServer: {

inline: true,

port: 8080

}, module: {

rules: [

{

test: /\.jsx?$/,

exclude: /node\_modules/,

loader: 'babel-loader',

query: {

presets: ['es2015', 'react']

}

}

]

},

plugins:[

new HtmlWebpackPlugin({template: './index.html'})

]

}

## Using the create-react-app command

Instead of using webpack and babel you can install ReactJS more simply by installing **create-react-app**.

### Step 1 - install create-react-app

Browse through the desktop and install the Create React App using command prompt as shown below −

D:\nodejs\_learning>npx create-react-app my-app

npx: installed 63 in 7.9s

Creating a new React app in D:\nodejs\_learning\my-app.

Installing packages. This might take a couple of minutes.

Installing react, react-dom, and react-scripts...

+ react-dom@16.8.3

+ react@16.8.3

+ [react-scripts@2.1](mailto:react-scripts@2.1).

added 1813 packages from 716 contributors and audited 36230 packages in 156.055s

found 63 low severity vulnerabilities

run `npm audit fix` to fix them, or `npm audit` for details

Success! Created my-app at D:\nodejs\_learning\my-app

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 my-app

npm start

Happy hacking!

This will create a folder named my-app and installs all the required files in it.

### Step 2 - Delete all the source files

Browse through the src folder in the generated my-app folder and remove all the files in it as shown below –

D:\nodejs\_learning>cd my-app

D:\nodejs\_learning\my-app>cd src

D:\nodejs\_learning\my-app\src>del \*

### Step 3 - Add files

Add files with names **index.css** and **index.js** in the src folder as –

In the index.js file add the following code

import React from 'react';

import ReactDOM from 'react-dom';

import './index.css';

### Step 4 - Run the project

Finally, run the project using the start command.

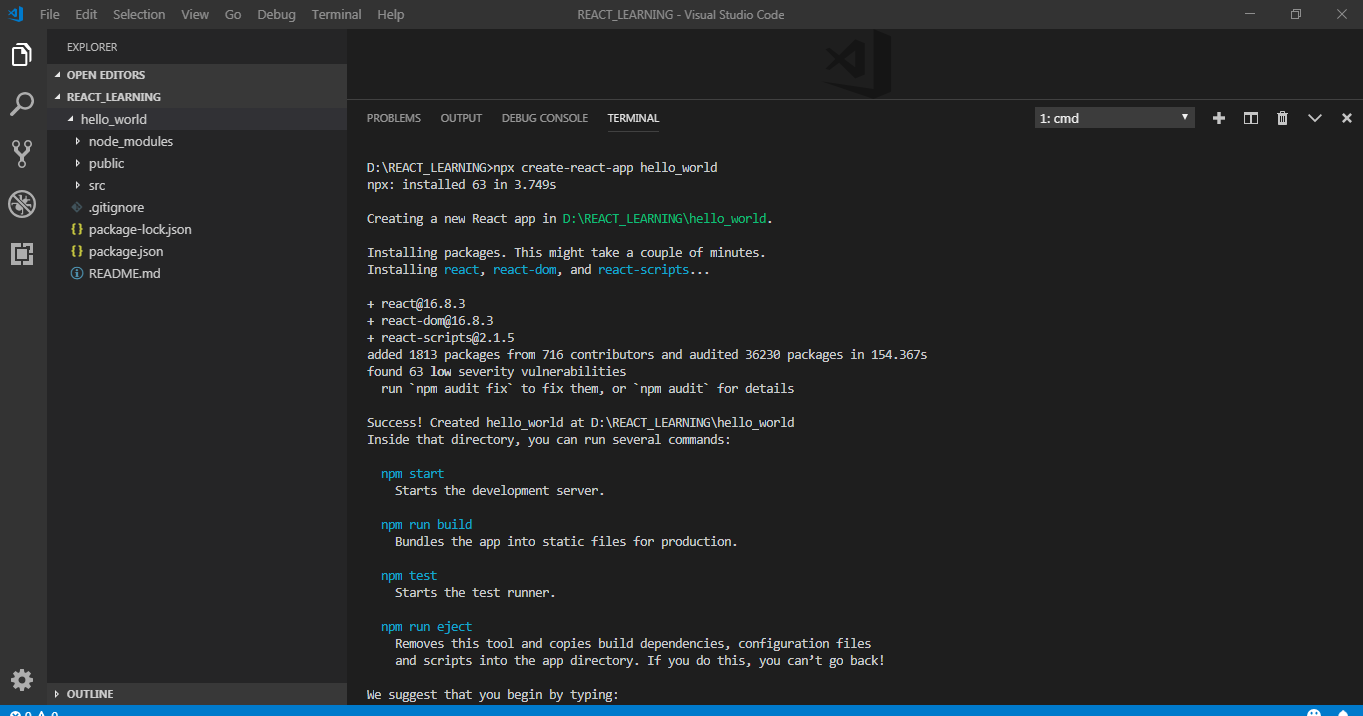
D:\nodejs\_learning\my-app>npm start

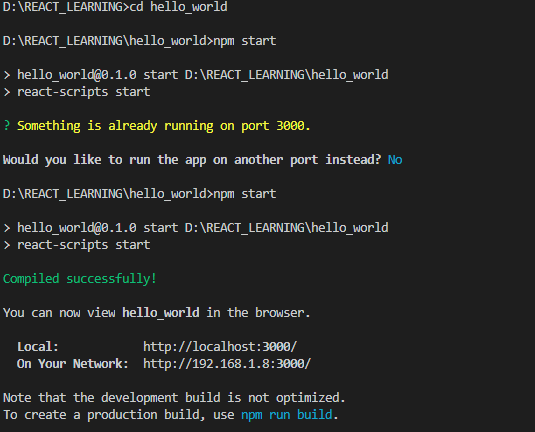
Using Visual Studio Code Editor

Create folder REACT\_LEARNING in any directory and open that folder in the terminal

D:\REACT\_LEARNING>

D:\REACT\_LEARNING>npx create-react-app hello\_world

go to that application and then start using npm start.



### 