

Welcome to Innominds Blog

Enjoy our insights and engage with us!

Create a react app without create-react-app

By Nabendu Kumar Biswas, February 9, 2021

[Tweet](#)[Share](#)[Like](#)[Share](#)

When it comes to start a react project, we all use the familiar create-react-app by Facebook engineers. This create-react-app stores the logic to use, Babel and Webpack behind the scene to make our React application work in the browser.

Did you know that you can also configure a React app, without using create-react-app? Here's how.

But first let's familiarize ourselves with we start Webpack and Babel.

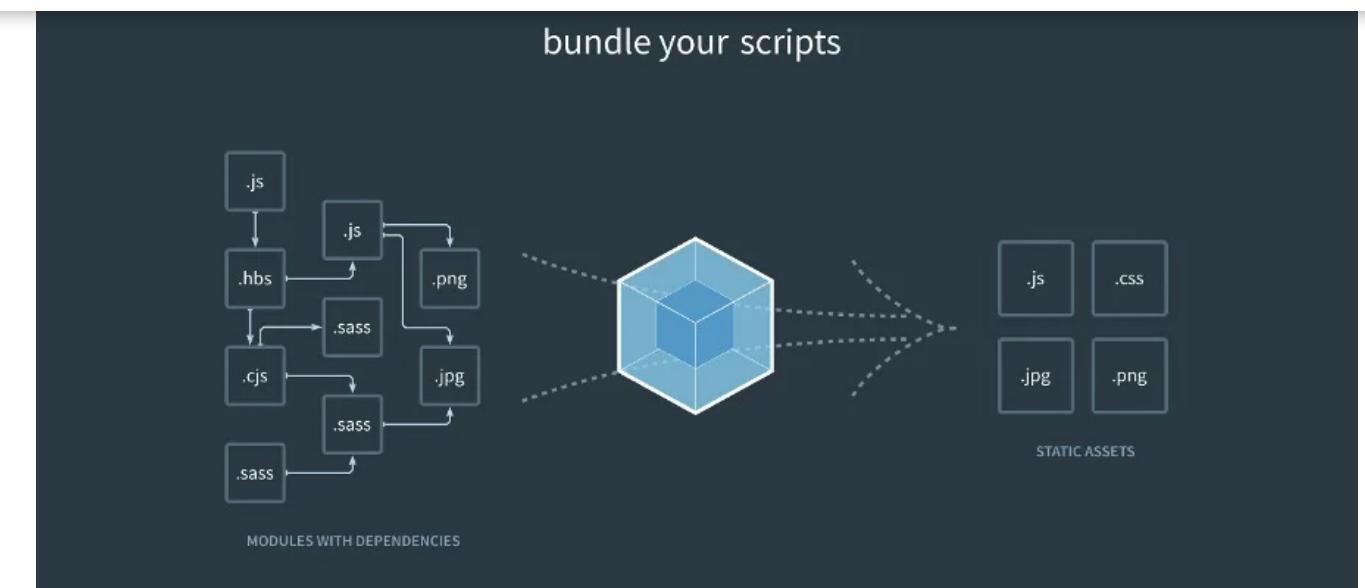
Webpack is a module builder. It should be noted that webpack doesn't run during your page load, but it runs during development.

Look at the below diagram from the website. It is self-explanatory on what webpack does. As shown on site, it takes your "js", "sass" and other dependencies files and converts them to plain "js", "css", "jpg" and "png" files which the browser can understand.

When we create an app with create-react-app, webpack is included in it and because of it in React we are able to use different components without worrying about the scope. And also our "scss", "jsx" files getting converted into "css" and "js".



Need any assistance?



Webpack

Babel is a JavaScript compiler that converts edge JavaScript (ES6) into plain old ES5 JavaScript that can run in any browser (even the old ones). In the React world it is used to transform the JSX code into vanilla JavaScript. The Babel website lists what you can do with Babel:

- Transform syntax
- Polyfill features that are missing in your target environment
- Source code transformations (codemods)

A simple example of what Babel does for arrow function.

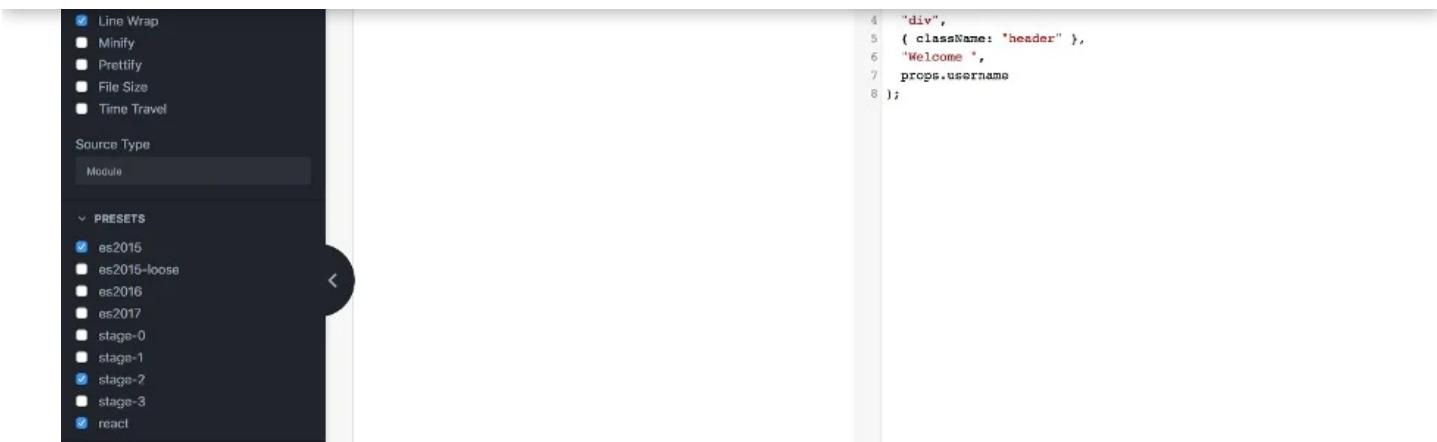
```
// Babel Input: ES2015 arrow function
[1, 2, 3].map((n) => n + 1); // Babel Output: ES5 equivalent
[1, 2, 3].map(function(n) {
  return n + 1;
});
```

Below is how a React JSX line gets converted into React createElement. In the first version of React, we used to write the confusing createElement way.

Need any assistance?

Innominds

Powering the Digital Next



Babel

Let's start our configuration now. We will first create a folder **webpack-react** and cd into it.

```

MINGW64/d/Projects/webpack > + -
pc@DESKTOP-3603RC5 MINGW64 /d/Projects
$ mkdir webpack-react
pc@DESKTOP-3603RC5 MINGW64 /d/Projects
$ cd webpack-react/
pc@DESKTOP-3603RC5 MINGW64 /d/Projects/webpack-react
$ |
  
```

webpack-react

Next, we will do **npm init** in the terminal, as all react app are basically NodeJS based. It will ask us some questions. We can leave most of them blank by pressing enter. We should keep the **package name** the same as the folder name and the rest are optional.

After the last option, you have to type **yes**.

Need any assistance?

Innominds

Powering the Digital Next

```
See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (webpack-react) webpack-react
version: (1.0.0)
description: React App without create-react-app
entry point: (index.js)
test command:
git repository:
keywords: [webpack-react]
author: Nabendu Biswas
license: (ISC)
About to write to D:\Projects\webpack-react\package.json:

[
  "name": "webpack-react",
  "version": "1.0.0",
  "description": "React App without create-react-app",
  "main": "index.js",
  "scripts": {
    "test": "echo \'Error: no test specified\' && exit 1"
  },
  "keywords": [
    "webpack-react"
  ],
  "author": "Nabendu Biswas",
  "license": "ISC"
}

Is this OK? (yes) yes
```

npm init

And it will create a **package.json** in our directory. Now, we need to install some npm packages, before moving forward. The first packages we are installing are for webpack itself by below command. Notice, that we are installing it as a dev dependency, which means it will not be installed in production.

npm i --save-dev webpack webpack-cli webpack-dev-server

```
pc@DESKTOP-3603RCS MINON64 /d/Projects/webpack-react
$ ls
package.json

pc@DESKTOP-3603RCS npm i --save-dev webpack webpack-cli webpack-dev-server
npm WARN deprecated chokidar@2.1.8: Chokidar 2 will break on node v14+. Upgrade to chokidar 3 with 15x less dependencies.
npm WARN deprecated fsevents@1.2.13: fsevents 1 will break on node v14+ and could be using insecure binaries. Upgrade to fsevents 2.
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.7 (node_modules\chokidar\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"})
npm WARN webpack-react@1.0.0 No repository field.

+ webpack-dev-server@3.11.1
+ webpack-cli@4.3.1
+ webpack@5.11.1
+ i@0.3.6
+ npm@6.14.10
added 942 packages from 1144 contributors and audited 943 packages in 35.64s

37 packages are looking for funding
  run `npm fund` for details

found 10 low severity vulnerabilities
  run `npm audit fix` to fix them, or `npm audit` for details

pc@DESKTOP-3603RCS MINON64 /d/Projects/webpack-react
$ |
```

Need any assistance?

webpack

```
npm i --save-dev eslint eslint-config-airbnb-base eslint-plugin-jest eslint-
config-prettier
```

```
pc@DESKTOP-3603RCS MINGW64 /d/Projects/webpack
+ -
pc@DESKTOP-3603RCS ~ % npm i --save-dev eslint eslint-config-airbnb-base eslint-plugin-jest eslint-config-prettier
npm WARN eslint-config-airbnb-base@14.2.1 requires a peer of eslint-plugin-import@^2.22.1 but none is installed. You must install peer dependencies yourself.
npm WARN tsutils@3.17.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.
npm WARN webpack-react@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
+ eslint-config-prettier@7.1.0
+ eslint-config-airbnb-base@14.2.1
+ eslint-plugin-jest@24.1.3
+ i@0.3.6
+ eslint@7.16.0
+ npm@6.14.10
added 108 packages from 77 contributors, updated 2 packages and audited 1051 packages in 13.286s

57 packages are looking for funding
  run `npm fund` for details

Found 10 low severity vulnerabilities
  run `npm audit fix` to fix them, or `npm audit` for details

pc@DESKTOP-3603RCS MINGW64 /d/Projects/webpack-react
$
```

eslint

Now, we are going to install a bunch of babel plugins. Babel is very important to convert ES6 into ES5, which is still required in old browsers. And also the jsx react code to createElement react code.

```
npm i --save-dev babel-loader @babel/preset-env @babel/core
@babel/plugin-syntax-dynamic-import @babel/plugin-transform-runtime
@babel/preset-react babel-eslint @babel/runtime
```

```
pc@DESKTOP-3603RCS MINGW64 /d/Projects/webpack-react
+ -
pc@DESKTOP-3603RCS ~ % npm i --save-dev babel-loader @babel/preset-env @babel/core @babel/plugin-syntax-dynamic-import @babel/plugin-transform-runtime @babel/preset-react babel-eslint @babel/runtime
npm WARN eslint-config-airbnb-base@14.2.1 requires a peer of eslint-plugin-import@^2.22.1 but none is installed. You must install peer dependencies yourself.
npm WARN tsutils@3.17.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.
npm WARN webpack-react@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
+ @babel/plugin-syntax-dynamic-import@7.0.0
+ @babel/plugin-transform-runtime@7.12.10
+ babel-eslint@10.1.0
+ @babel/runtime@7.12.5
+ @babel/core@7.12.10
+ babel-loader@8.2.2
+ @babel/preset-env@7.12.10
+ @babel/preset-env@7.12.11
added 142 packages from 69 contributors and audited 1190 packages in 8.684s

57 packages are looking for funding
  run `npm fund` for details

Found 38 low severity vulnerabilities
  run `npm audit fix` to fix them, or `npm audit` for details

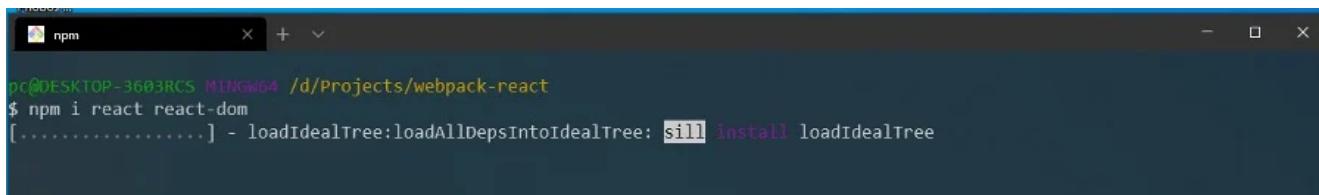
pc@DESKTOP-3603RCS MINGW64 /d/Projects/webpack-react
$
```

Need any assistance?

Innominds

Powering the Digital Next

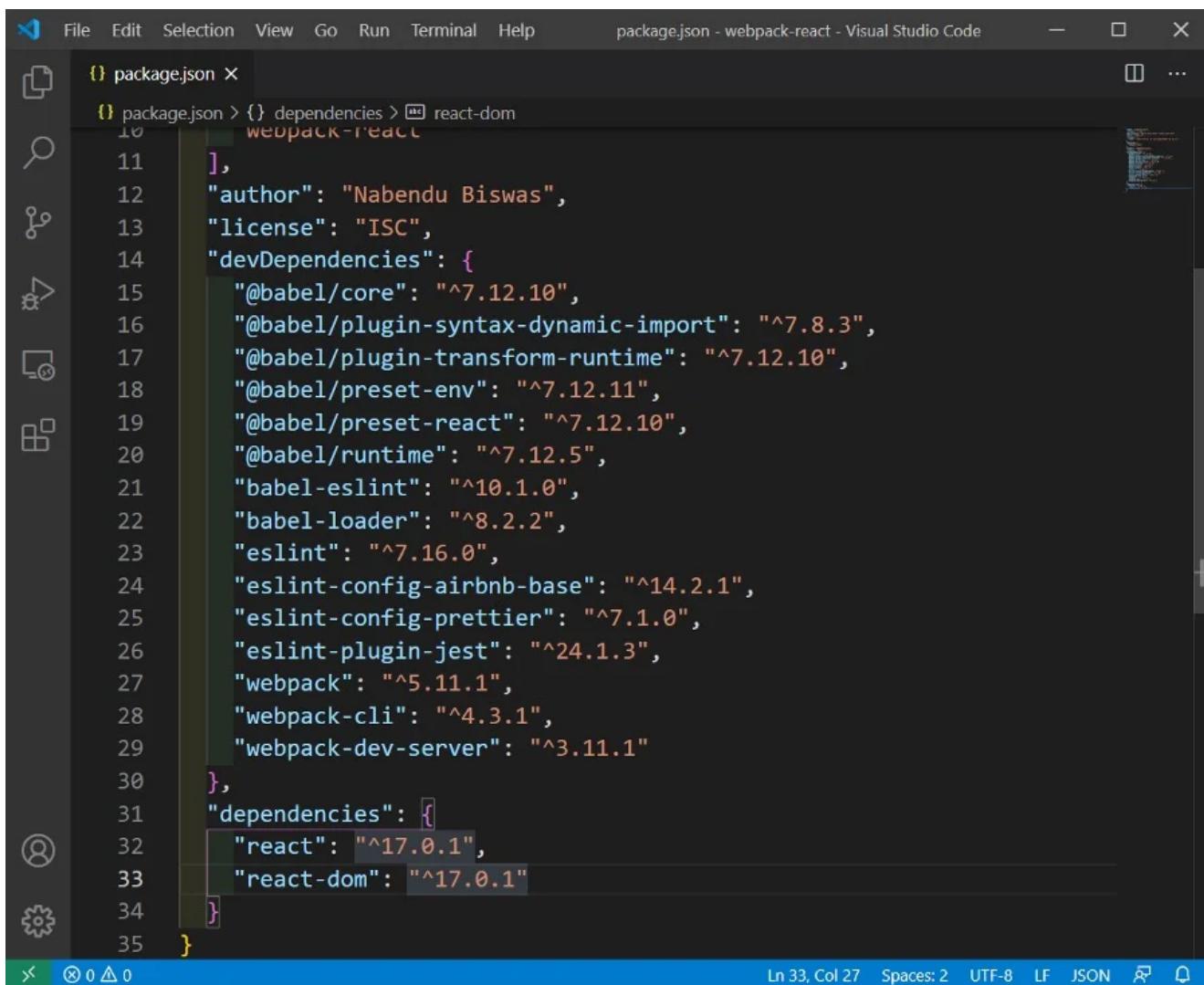
<https://www.innominds.com/react/react-dom>



```
pc@DESKTOP-3603RCS MINGW64 /d/Projects/webpack-react
$ npm i react react-dom
[.....] - loadIdealTree:loadAllDepsIntoIdealTree: sill install loadIdealTree
```

react

Now, we will open our project in VS Code, we can see all the dev dependencies and dependencies in the **package.json** file.



```
{
  "name": "webpack-react",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "start": "node index.js"
  },
  "dependencies": {
    "react": "^17.0.1",
    "react-dom": "^17.0.1"
  },
  "devDependencies": {
    "@babel/core": "^7.12.10",
    "@babel/plugin-syntax-dynamic-import": "^7.8.3",
    "@babel/plugin-transform-runtime": "^7.12.10",
    "@babel/preset-env": "^7.12.11",
    "@babel/preset-react": "^7.12.10",
    "@babel/runtime": "^7.12.5",
    "babel-eslint": "^10.1.0",
    "babel-loader": "^8.2.2",
    "eslint": "^7.16.0",
    "eslint-config-airbnb-base": "^14.2.1",
    "eslint-config-prettier": "^7.1.0",
    "eslint-plugin-jest": "^24.1.3",
    "webpack": "^5.11.1",
    "webpack-cli": "^4.3.1",
    "webpack-dev-server": "^3.11.1"
  }
}
```

package.json

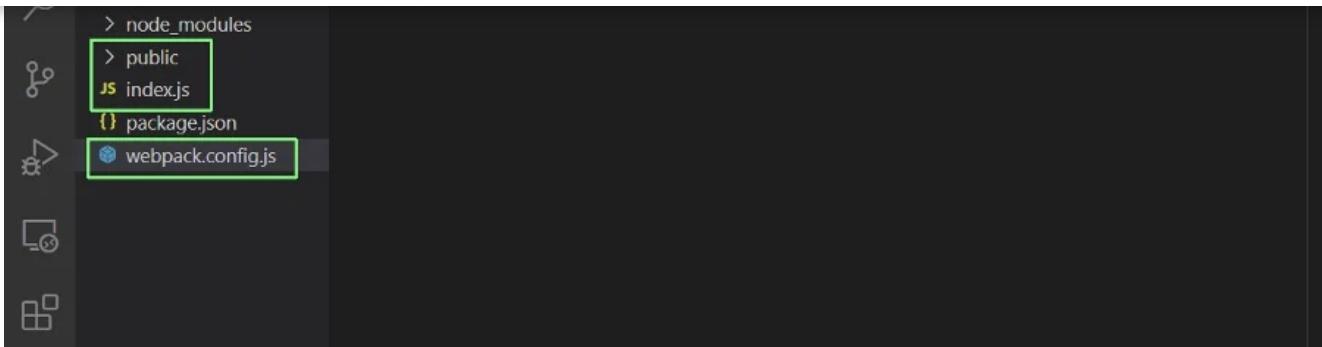
Now, we need to create a **webpack.config.js** file in the same directory. This file basically tells us how to bundle the complete project in a simple f

Need any assistance?

We will also create **public** folder and a **index.js** file in the same directory.

Innominds

Powering the Digital Next



configuration

We will now start to write our webpack configuration in **webpack.config.js** file. Here, inside module.exports we are first telling the mode as development. After that the entry point of our project which is **index.js**.

Next, the output of our project which will be **main.js** inside the **public** folder. To get the relative path, we will be importing the path at line 1 and also using it within our output object.

We are basically telling, webpack take all the things that are present in **index.js**, along with the dependencies that are imported in index.js and produce a file **main.js** and put it in **public** folder.

```

const path = require("path");

module.exports = {
  mode: "development",
  entry: "./index.js",
  output: {
    path: path.resolve(__dirname, "public"),
    filename: "main.js"
  }
}

```

webpack.config.js

Next, we will first mention that the **target** is a node application. After that we will put configuration for **devServer**. Here, we can mention any port for our application to run then the content is from public folder.

We are also resolving few file **extensions**, as we need to **it needs to take** to produce the final output.

Need any assistance? [Get Help](#)

Innominds

Powering the Digital Next

```

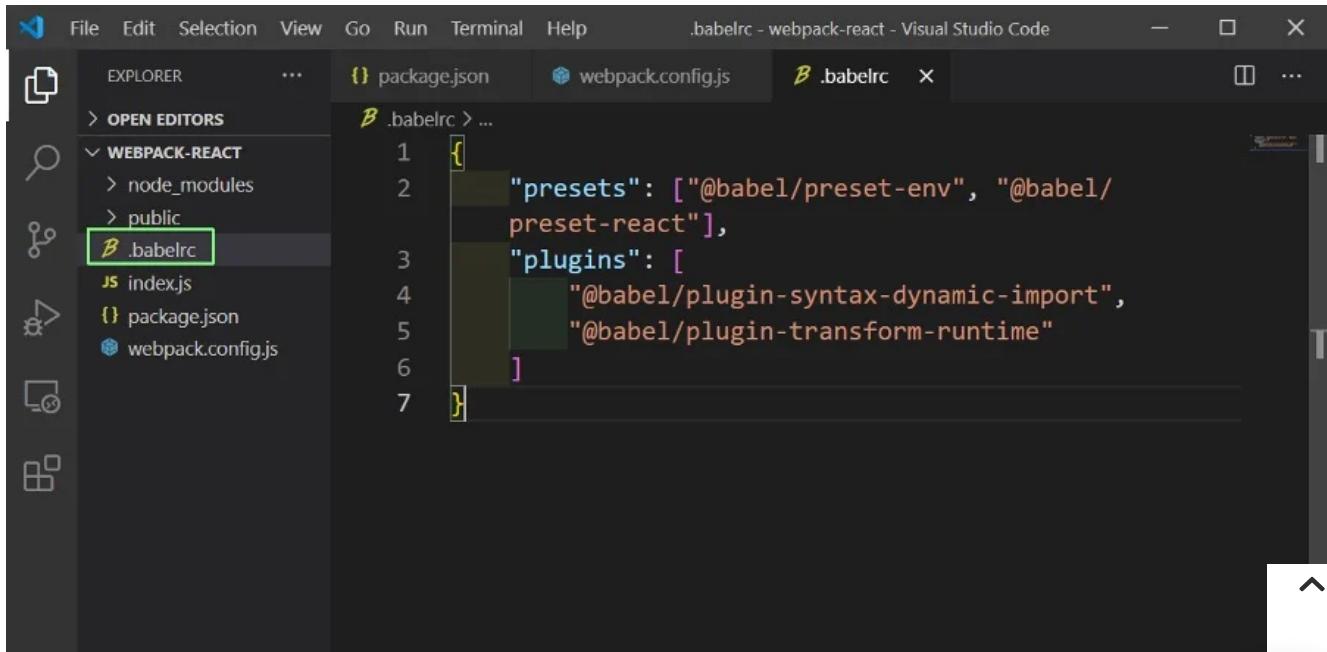
3 module.exports = [
4   mode: "development",
5   entry: "./index.js",
6   output: {
7     path: path.resolve(__dirname, "public"),
8     filename: "main.js"
9   },
10  target: "node",
11  devServer: {
12    port: "9500",
13    contentBase: ["./public"],
14    open: true
15  },
16  resolve: {
17    extensions: [".js", ".jsx", ".json"]
18  },
19]

```

webpack.config.js

We also need a babel configuration file, before creating some rules in our **webpack.config.js** file. So, create a file **.babelrc** in the root directory and the below content in it.

It is a sample babel configuration which allows us to use configuration of presets and some plugins for the transformation.



```

presets: ["@babel/preset-env", "@babel/preset-react"],
plugins: [
  "@babel/plugin-syntax-dynamic-import",
  "@babel/plugin-transform-runtime"
]

```

.babelrc

Need any assistance?

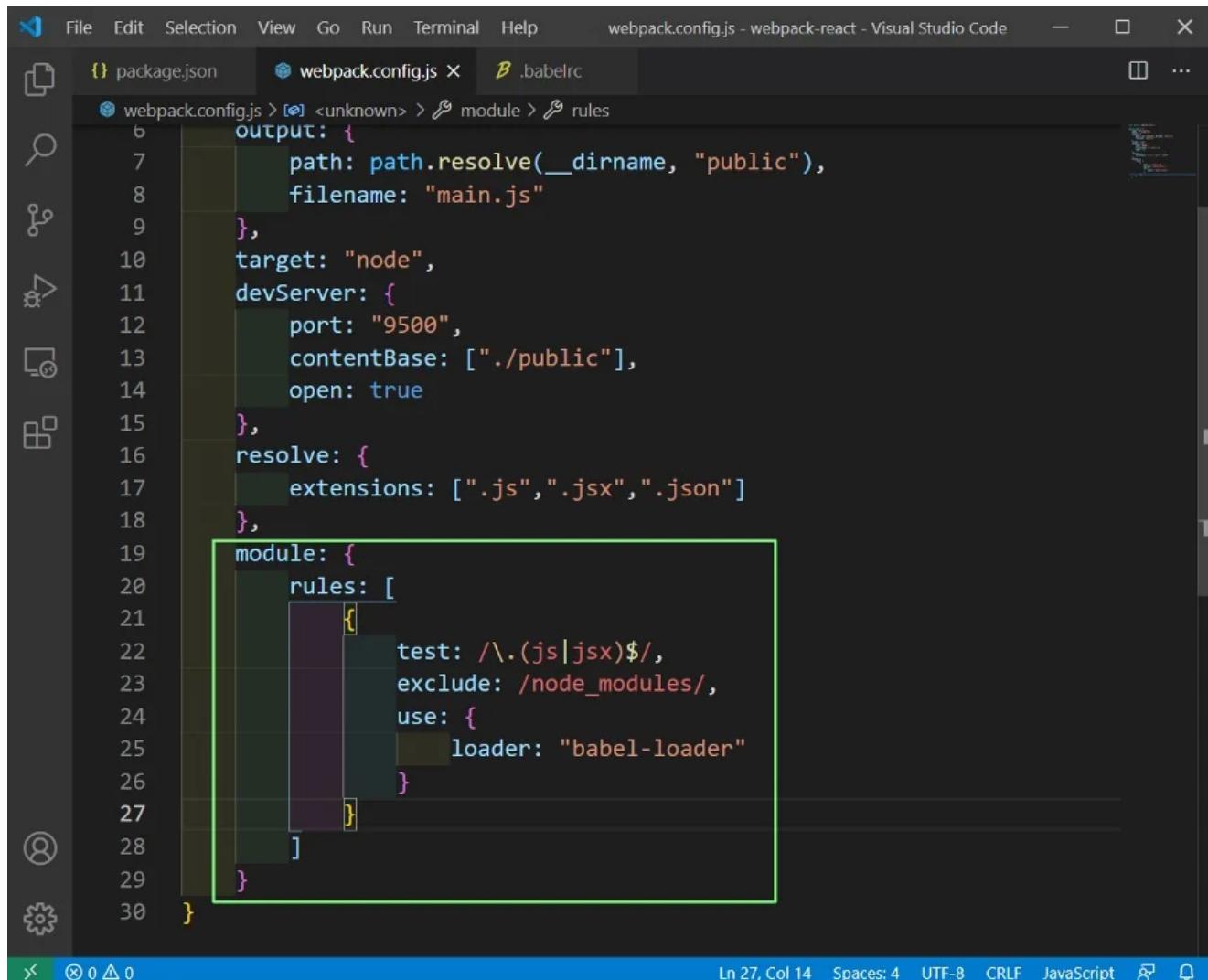
Innominds

Powering the Digital Next

whose value is a regular expression which gives us js or jsx files.

Next, we want our **node_modules** folder to be excluded of these rules because the creator of every npm package has already applied these rules.

Now, we need to mention the loader which we are going to use. Here it is the **babel-loader**.



```

File Edit Selection View Go Run Terminal Help webpack.config.js - webpack-react - Visual Studio Code
package.json webpack.config.js .babelrc

webpack.config.js > <unknown> > module > rules
  output: {
    path: path.resolve(__dirname, "public"),
    filename: "main.js"
  },
  target: "node",
  devServer: {
    port: "9500",
    contentBase: ["./public"],
    open: true
  },
  resolve: {
    extensions: [".js", ".jsx", ".json"]
  },
  module: {
    rules: [
      {
        test: /\.js|jsx$/,
        exclude: /node_modules/,
        use: {
          loader: "babel-loader"
        }
      }
    ]
  }
}
Ln 27, Col 14 Spaces: 4 UTF-8 CRLF JavaScript

```

webpack.config.js

Our **webpack.config.js** file is now complete. Let's again understand what this file does.

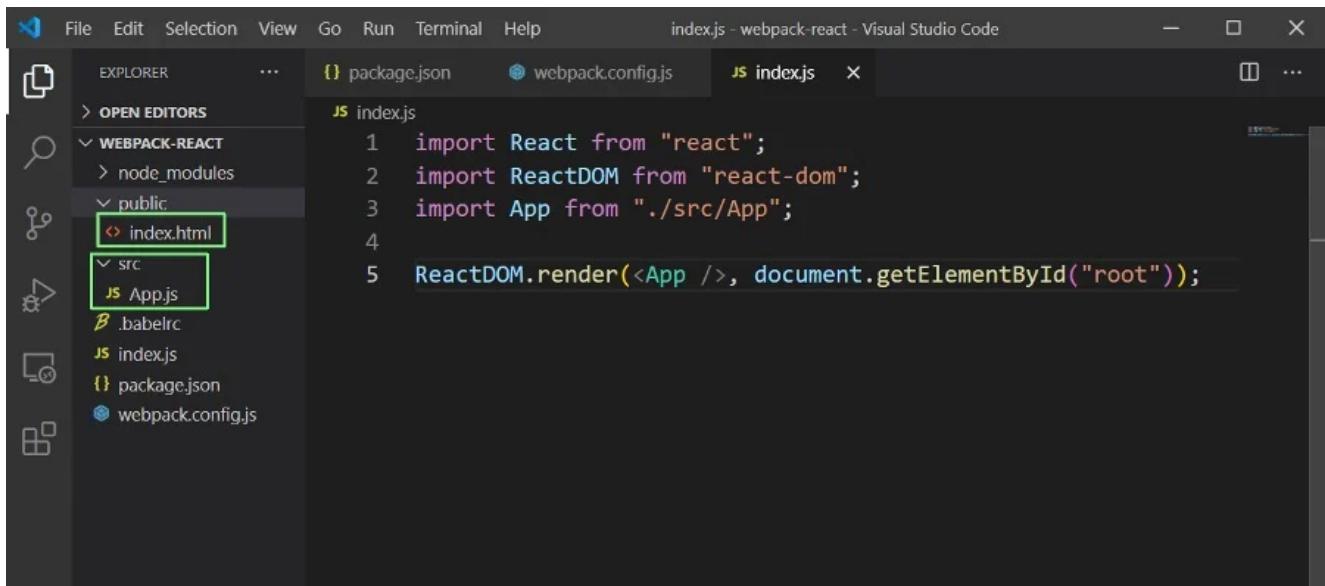
We are basically telling webpack to take index.js from **entry**. Then check for all file extensions in **resolve**. After that apply all the **rules** in **module.rules** and produce the **output** and place it in main.js in the **public** folder.

Next, create a file **App.js** inside a newly created **src** folder and **index.html** inside the **public** folder. Our folder structure is looking a lot similar. Need any assistance? [us.](#)

Innominds

Powering the Digital Next

Notice that we are using `document.getElementById` from JavaScript DOM.



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure:
 - WEBPACK-REACT** folder
 - node_modules**
 - public** folder
 - index.html** (highlighted with a red box)
 - src** folder
 - App.js** (highlighted with a red box)
 - index.js** file (highlighted with a red box) contains the following code:

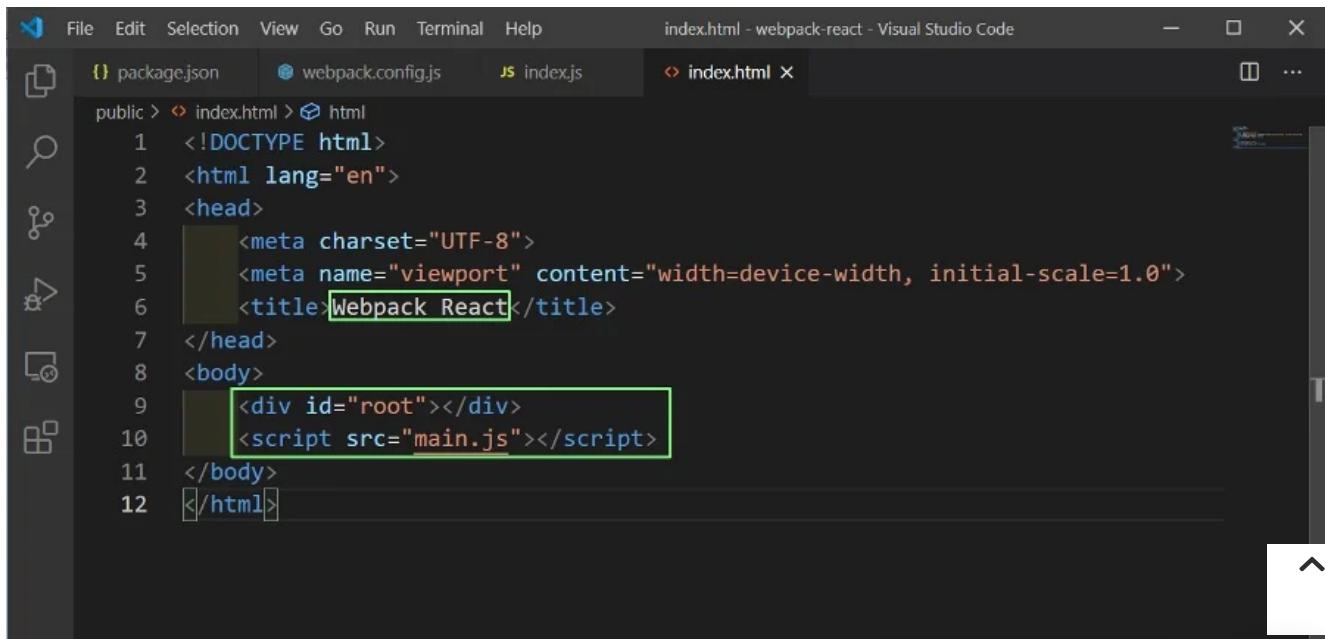

```
1 import React from "react";
2 import ReactDOM from "react-dom";
3 import App from "./src/App";
4
5 ReactDOM.render(<App />, document.getElementById("root"));
```

index.js

Next, we are going to create this root element in **index.html** file. First generate the html boilerplate by ! and pressing tab.

After that change the title and create a single div with id of root. Now, we will also create a script which will load main.js file.

We still don't have the **main.js** because it will be created by webpack which will convert all of this complex react code to javascript.



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure:
 - public** folder
 - index.html** (highlighted with a red box)

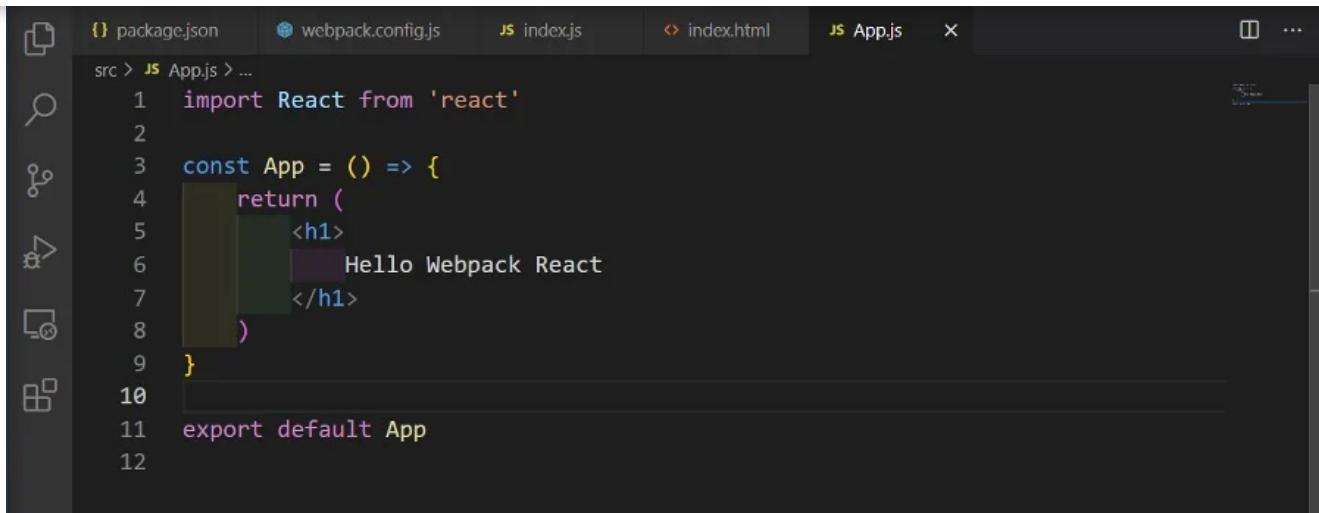

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Webpack React</title>
7 </head>
8 <body>
9   <div id="root"></div>
10  <script src="main.js"></script>
11 </body>
12 </html>
```

index.html

Need any assistance?

Innominds

Powering the Digital Next



```

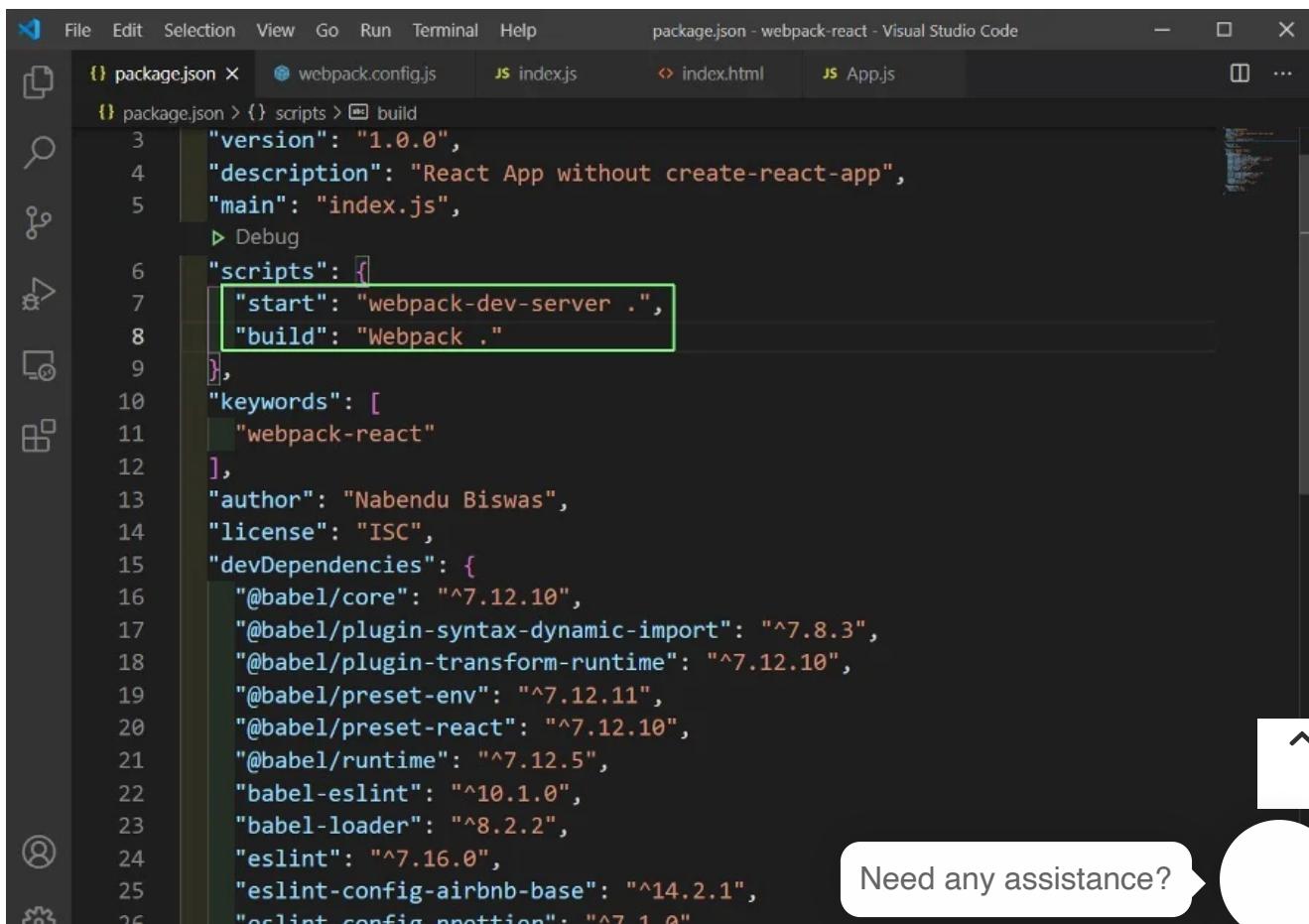
package.json webpack.config.js index.js index.html App.js
src > JS App.js ...
1 import React from 'react'
2
3 const App = () => {
4   return (
5     <h1>
6       Hello Webpack React
7     </h1>
8   )
9 }
10
11 export default App
12

```

App.js

Before testing our project, we need some node scripts in our **package.json** file. The **start** script asks to run the webpack-dev-server in our current project at port 9500, from the public folder.

The build command asks us to **build** this package in main.js file. It actually runs all logic in **webpack.config.js** file.



```

File Edit Selection View Go Run Terminal Help package.json - webpack-react - Visual Studio Code
File Edit Selection View Go Run Terminal Help package.json - webpack-react - Visual Studio Code
package.json webpack.config.js index.js index.html App.js
{
  "version": "1.0.0",
  "description": "React App without create-react-app",
  "main": "index.js",
  "scripts": {
    "start": "webpack-dev-server .",
    "build": "Webpack ."
  },
  "keywords": [
    "webpack-react"
  ],
  "author": "Nabendu Biswas",
  "license": "ISC",
  "devDependencies": {
    "@babel/core": "^7.12.10",
    "@babel/plugin-syntax-dynamic-import": "^7.8.3",
    "@babel/plugin-transform-runtime": "^7.12.10",
    "@babel/preset-env": "^7.12.11",
    "@babel/preset-react": "^7.12.10",
    "@babel/runtime": "^7.12.5",
    "babel-eslint": "^10.1.0",
    "babel-loader": "^8.2.2",
    "eslint": "^7.16.0",
    "eslint-config-airbnb-base": "^14.2.1",
    "eslint-config-react-native": "^7.1.0"
  }
}

```

Need any assistance?

Innominds

Powering the Digital Next

```
pc@DESKTOP-3603RCS MINGW64 /d/Projects/webpack-react
$ npm run build

> webpack-react@1.0.0 build D:\Projects\webpack-react
> Webpack.

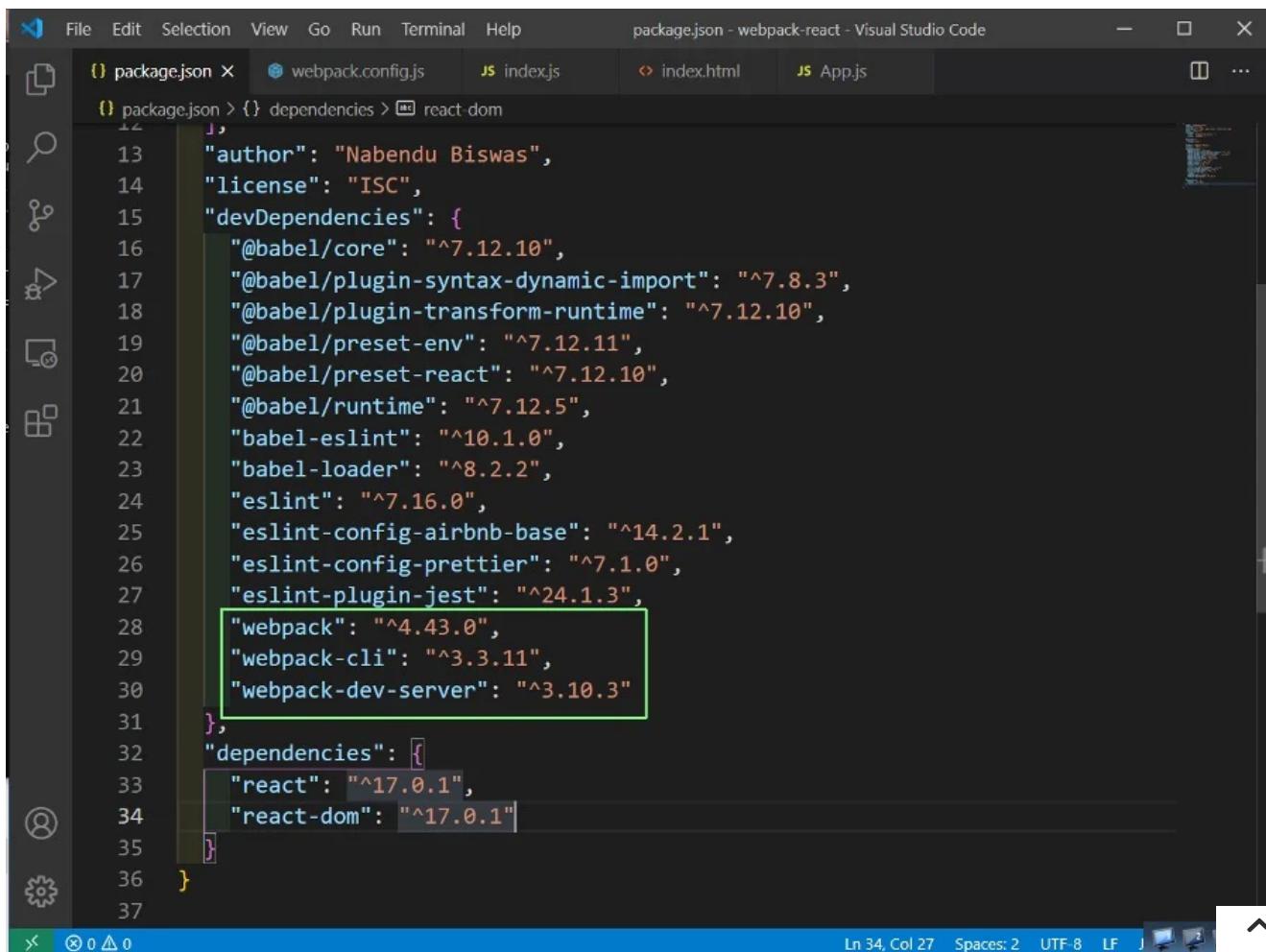
[webpack-cli] Unknown command .
[webpack-cli] Run 'webpack --help' to see available commands and options
npm ERR! code ELIFECYCLE
npm ERR! errno 2
npm ERR! webpack-react@1.0.0 build: `Webpack .`
npm ERR! Exit status 2
npm ERR!
npm ERR! Failed at the webpack-react@1.0.0 build script.
npm ERR! This is probably not a problem with npm. There is likely additional logging output above.

npm ERR! A complete log of this run can be found in:
npm ERR!     C:\Users\pc\AppData\Roaming\npm-cache\_logs\2021-01-01T13_55_44_151Z-debug.log

pc@DESKTOP-3603RCS MINGW64 /d/Projects/webpack-react
```

npm run build

Actually, there is an issue at the time of writing this post with current version of webpack. So, we will go back to the earlier version by changing it in **package.json** file.



```
File Edit Selection View Go Run Terminal Help package.json - webpack-react - Visual Studio Code
package.json X webpack.config.js index.js index.html App.js

{
  "name": "webpack-react",
  "version": "1.0.0",
  "author": "Nabendu Biswas",
  "license": "ISC",
  "devDependencies": {
    "@babel/core": "^7.12.10",
    "@babel/plugin-syntax-dynamic-import": "^7.8.3",
    "@babel/plugin-transform-runtime": "^7.12.10",
    "@babel/preset-env": "^7.12.11",
    "@babel/preset-react": "^7.12.10",
    "@babel/runtime": "^7.12.5",
    "babel-eslint": "^10.1.0",
    "babel-loader": "^8.2.2",
    "eslint": "^7.16.0",
    "eslint-config-airbnb-base": "^14.2.1",
    "eslint-config-prettier": "^7.1.0",
    "eslint-plugin-jest": "^24.1.3",
    "webpack": "^4.43.0",
    "webpack-cli": "^3.3.11",
    "webpack-dev-server": "^3.10.3"
  },
  "dependencies": {
    "react": "^17.0.1",
    "react-dom": "^17.0.1"
  }
}
```

package.json

To install these new packages, we have to do **npm i** in the Need any assistance? a packages from package.json file in our project.

Innominds

Powering the Digital Next

```

npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@2.1.3: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@^1.2.7 (node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"})
npm WARN eslint-config-airbnb-base@14.2.1 requires a peer of eslint-plugin-import@^2.22.1 but none is installed. You must install peer dependencies yourself.
npm WARN tsutils@3.17.1 requires a peer of typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta but none is installed. You must install peer dependencies yourself.
npm WARN webpack-react@1.0.0 No repository field.

added 138 packages from 102 contributors, removed 29 packages, updated 39 packages and audited 872 packages in 10.831s

54 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

pc@DESKTOP-3603RCS MINGW64 /d/Projects/webpack-react
$
```

npm i

Now, our **npm run build** is a success.

```

pc@DESKTOP-3603RCS MINGW64 /d/Projects/webpack-react
$ npm run build

> webpack-react@1.0.0 build D:\Projects\webpack-react
> Webpack .

Hash: 45870078ab993ffd8326
Version: webpack 4.44.2
Time: 1349ms
Built at: 01/01/2021 7:31:39 PM
    Asset      Size  Chunks      Chunk Names
  main.js  1020 KiB  main  [emitted]  main
Entrypoint main = main.js
[./index.js] 185 bytes {main} [built]
[./src/App.js] 156 bytes {main} [built]
  + 9 hidden modules

pc@DESKTOP-3603RCS MINGW64 /d/Projects/webpack-react
$
```

npm run build

And it generates this huge **main.js** file in our **public** folder. The file is actually **1019 KB (1 MB)** in size. This is our development build.

Need any assistance?

Innominds

Powering the Digital Next

```

2  // The module cache
3  var installedModules = {};
4
5  // The require function
6  function _webpack_require_(moduleId) {
7
8    // Check if module is in cache
9    if(installedModules[moduleId]) {
10      return installedModules[moduleId].exports;
11    }
12    // Create a new module (and put it into the
13    // cache)
14    var module = installedModules[moduleId] = {
15      i: moduleId,
16      l: false,
17      exports: {}
18    };
19    // Execute the module function
20    modules[moduleId].call(module.exports, module,
21      module.exports, _webpack_require_);
22    // Flag the module as loaded
23    module.l = true;
24

```

main.js

Now, we will see whether our application will run, by giving **npm start** from terminal.

Giving the command starts our dev server in <http://localhost:9500/>

```

$ npm start
> webpack-react@1.0.0 start D:\Projects\webpack-react
> webpack-dev-server .
[...]
Project is running at http://localhost:9500/
webpack output is served from /
Content not from webpack is served from ./public
Hash: adedc8a1a03d4f1c3c3f7
Version: webpack 4.44.2
Time: 15ms
Built at: 01/01/2021 7:38:53 PM
Asset Size Chunks Chunk Names
main.js 1020 KiB main [emitted] main
Entrypoint main = main.js
[0] multi ./28 bytes {main} [built]
[1] ./index.js 185 bytes {main} [built]
[2] node_modules/object-assign/index.js 2.06 KiB {main} [built]
[3] node_modules/react/cjs/react.development.js 874 KiB {main} [built]
[4] node_modules/react-dom/cjs/react-dom.development.js 1.71 MiB {main} [built]
[5] node_modules/react/cjs/react.development.js 70.5 KiB {main} [built]
[6] node_modules/react/index.js 106 bytes {main} [built]
[7] node_modules/scheduler/cjs/scheduler-tracing.development.js 8.79 KiB {main} [built]
[8] node_modules/scheduler/cjs/scheduler.development.js 22.6 KiB {main} [built]
[9] node_modules/scheduler/tracing.js 198 bytes {main} [built]
[10] ./src/App.js 156 bytes {main} [built]
[11] ./index.html compiled successfully.

```

npm start

We will convert this into a **production** build by changing the mode to production in **webpack.config.js** file. We have also deleted the **main.js** file from **public** folder.

Need any assistance?

Innominds

Powering the Digital Next

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure:
 - > node_modules
 - public
 - index.html
 - src
 - App.js
 - .babelrc
 - index.js
 - { package-lock.json
 - { package.json
 - webpack.config.js

- Code Editor (Right):** Displays the content of `webpack.config.js`. The code defines a module.exports object with various configuration properties like mode, entry, output, target, devServer, resolve, and module rules.
- Bottom Status Bar:** Shows the current file is `webpack.config.js`, with line 14, column 19, and other status indicators like UTF-8, CRLF, JavaS, and browser icons.

webpack.config.js

Run the `npm run build` command in the terminal to generate a new `main.js` file once again.

```
pc@DESKTOP-3603RCS MINGW64 /d/Projects/webpack-react
$ npm run build

> webpack-react@1.0.0 build D:\Projects\webpack-react
> Webpack .

Hash: 105d73795ef579c45dc
Version: webpack 4.44.2
Time: 2084ms
Built at: 01/01/2021 7:43:32 PM
Asset      Size  Chunks      Chunk Names
main.js    129 KiB   0  [emitted]  main
Entrypoint main = main.js
[7] ./index.js + 1 modules 351 bytes {0} [built]
|  ./index.js 185 bytes [built]
|  ./src/App.js 156 bytes [built]
+ 7 hidden modules
[1] ./node_modules/react-dom/index.js 1.1 MiB [built]
[2] ./node_modules/react/index.js 1.1 MiB [built]
[3] ./node_modules/react-dom/client/index.js 1.1 MiB [built]
[4] ./node_modules/react-dom/client/index.js.map 1.1 MiB [built]
[5] ./node_modules/react/index.js.map 1.1 MiB [built]
[6] ./node_modules/react-dom/client/index.js.map 1.1 MiB [built]

pc@DESKTOP-3603RCS MINGW64 /d/Projects/webpack-react
$
```

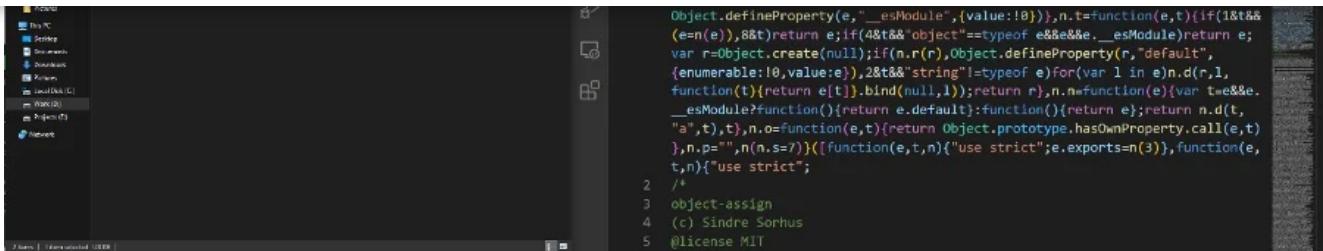
build

If we check the size of `main.js` file it's only **129 KB** and the file is also completely minified.

Need any assistance?

Innominds

Powering the Digital Next



main.js

With the optimization from **1019 KB** to **129 KB**, we might want to use production build always. But, we should use development mode while doing development because the hot reloading is faster in development mode.

This completes our easy way to create a react app. Hope, you guys use it in your next build instead of create-react-app.

The code for the same is in [this](#) github repo.

Topics: App Development



Nabendu Kumar Biswas
Lead - Software Engineering

Name*

Email*

Comment*

Need any assistance?