**Step 1: create package.json**

Using npm init command in your terminal

>npm init

**Step 2: Add details for** **package.json**

After hitting command it will ask following details needs to be filled, if you want to leave any field leave empty just hit enter.

This utility will walk you through creating a package.json file.

It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` 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: (bilor\_plate)

version: (1.0.0)

description: Boilerplate for my react webpack application

entry point: (index.js)

test command:

git repository:

keywords:

author: Narenkrishna

license: (ISC)

About to write to D:\React\Webpack\Bilor\_plate\package.json:

{

"name": "bilor\_plate",

"version": "1.0.0",

"description": "Boilerplate for my react webpack application",

"main": "index.js",

"scripts": {

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

},

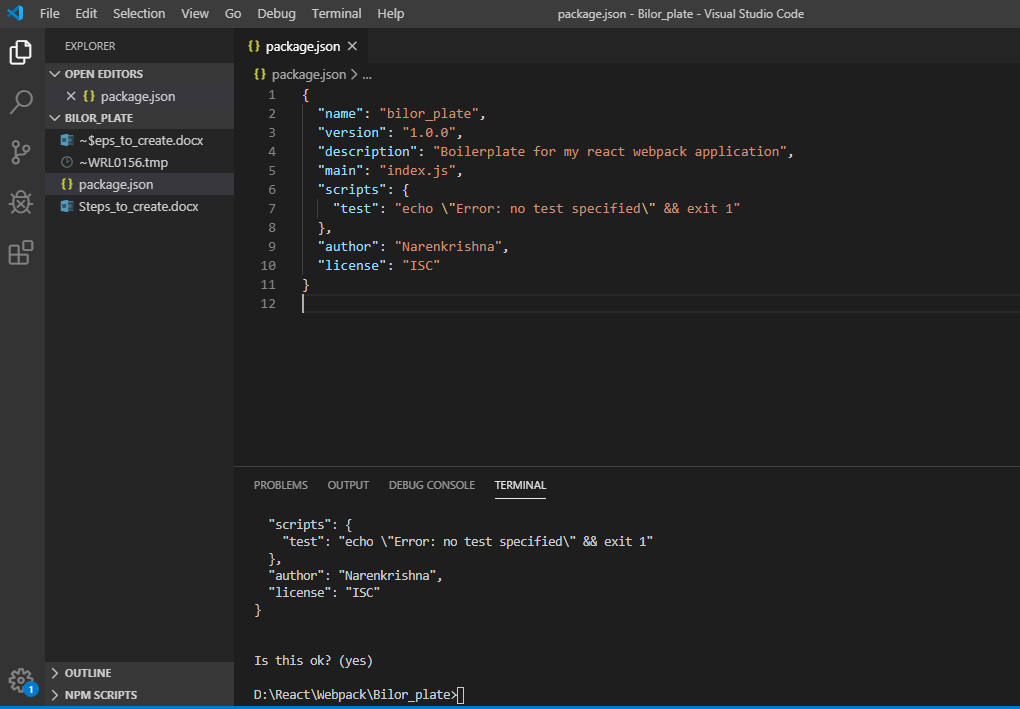
"author": "Narenkrishna",

"license": "ISC"

}

Is this ok? (yes)

After this the package.json will created in your application, like this.



**Step 3: Install react**

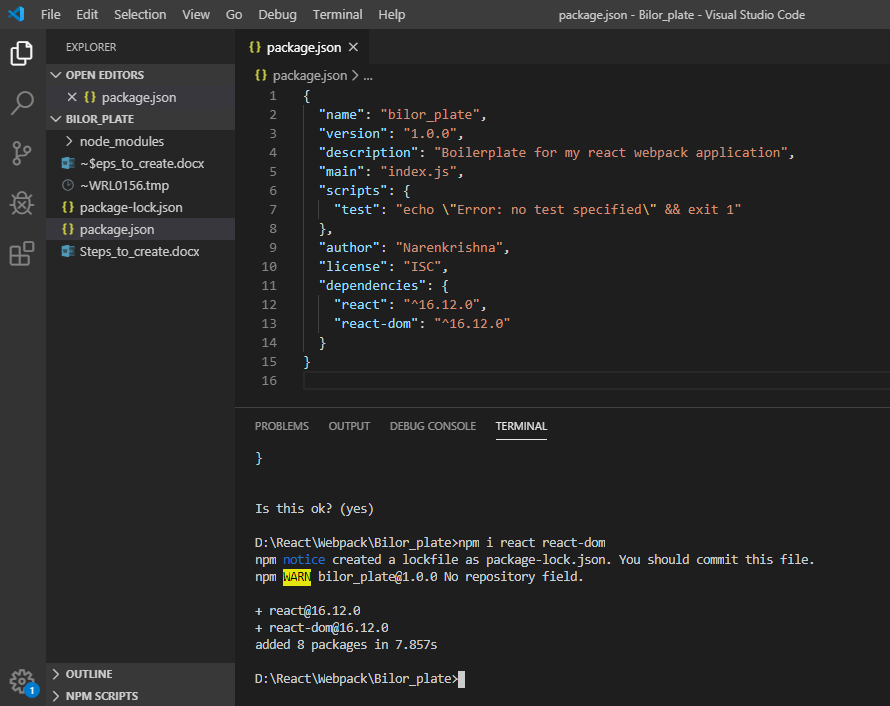
Now this is time to install react. Basically react has two parts such as react and react-dom. So now place below command in your terminal.

* npm i react react-dom

or

npm install react react-dom

Now the react library will added to your package.json under “dependencies”, like this

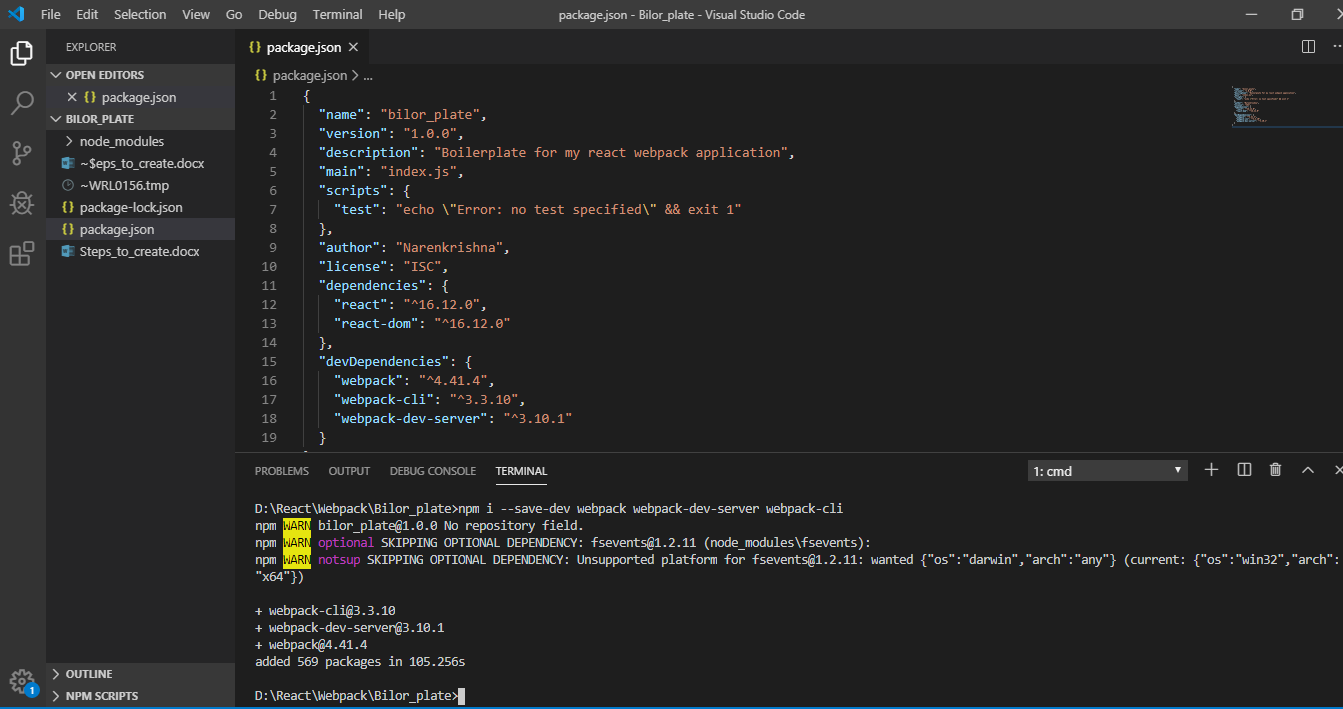


**Step 4: Install webpack**

Now we are going to install devdependencies. In webpack in react all our react codes are component has to be complied bundled into one JavaScript file through the webpack concern. Also we have to install dev server called webpack-dev-server and webpack cli which allows us to run webpack commands actually uses for build script. So now follow these commands to install webpack, webpack dev server and webpack-cli.

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

Now the webpack, webpack dev server and webpack-cli added to your package.json under “devdependencies”, like this



Step 5: **Install babel**

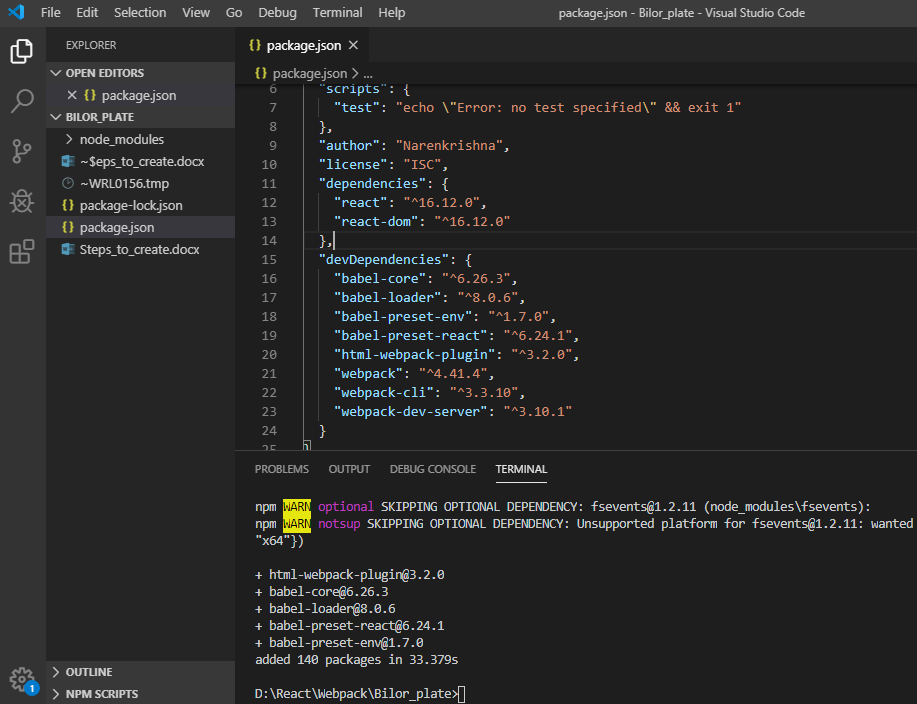
**Babel** is a JavaScript compiler that includes the ability to compile JSX into regular JavaScript. Babel we need because uses ES6 classes imports stuff like that. So we need to transpile the code to browser friendly code depending on which browser has been used. And also we need babel along with two plugins are presets. One is actually called **babel-preset-react** and other one called **babel-preset-env**. Because the env preset does only compile es6 but also later versions and it actually what browser being used and whatever needs to be complied for browsers.

Also we need loader called **babel-loader**. Because we needs to be able to compile jsx which is basically react uses for tempering. Also we goanna use **html-webpack-plugin** which will generate a build html file.

So use these commands for above specifications.

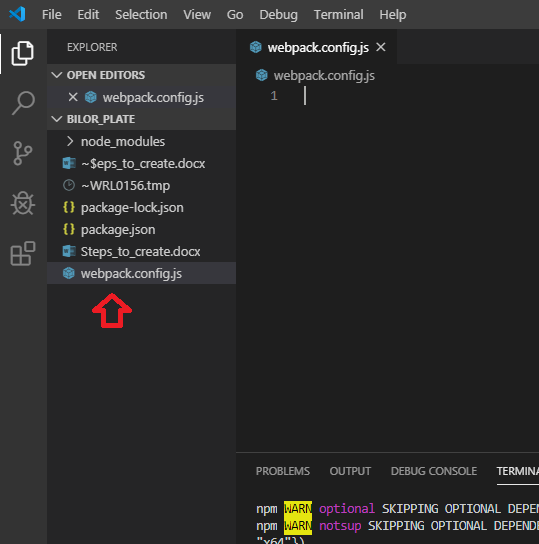
* npm i --save-dev babel-core babel-loader babel-preset-env babel-preset-react html-webpack-plugin

Now the babel-core, babel-loader, babel-preset-env, babel-preset-react and html-webpack-plugin added to your package.json under “devdependencies”, like this



**Step 6: Setup webpack config**

1. First create webpack.config.js in your root folder.



1. Then create path and HtmlWebpackPlugin.

const path = require('path');

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

1. Now create module with object here we are using former syntax module.exports = {}

module.exports = {

}

1. Add entry point on that object. It should be in our “./src/index.js” so create “src” folder under the root package and create index.js under src.

module.exports = {

    entry:'./src/index.js',

}

1. Add output. Which has our compile code has to go in one JavaScript file its goanna be a bundled file. Basically this is object in syntax. So we need specify two things path to join current directory and the output folder. And Specify the file name we can give name here “example: bundle.js”.

module.exports = {

    entry:'./src/index.js',

    output:{

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

filename: 'bundle.js'

    }

}

1. Next we are goanna add loader under the module object and in which under the rules array. Basically it is a regular expression put it in a **“test”** key. And make sure to exclude the **node\_modules** same. And added **babel-loader** inside the **“use”** object.

module.exports = {

    entry:'./src/index.js',

    output:{

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

        filename: 'bundle.js'

    },

    module:{

        rules:[

            {

                test: /\.js$/,

                exclude: /node\_modules/,

                use:{

                    loader: 'babel-loader'

                }

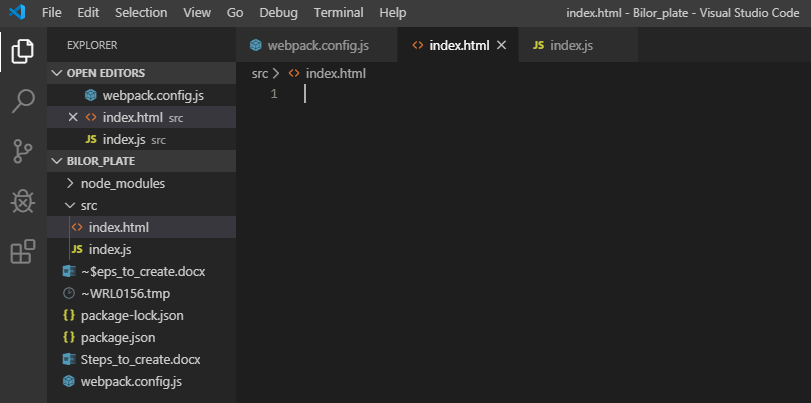
            }

        ]

    }

}

1. Html webpack plugin basically create an index.html file while build the application to our target folder. For this first we have to create **index.html** inside the **“src”**. This plugin that simplifies creation of HTML files to serve your bundles. This can be done by build command. This index.html will add react application. Here react dom will focus the id.



1. Now add the basic code inside the html file. Here we no need to worry about to add the bundle.js file inside the script tag webpack will do for us.

Ref: <https://www.google.com/search?sxsrf=ACYBGNSKqZRkrgzhy5ERNVoSRoC-bSDrAg%3A1577341793776&ei=YVMEXriPL-Wa4-EPr-aMmA4&q=html-webpack-plugin&oq=html-webpack-plugin&gs_l=psy-ab.3..0l5j0i7i30j0j0i7i30l3.26030.28613..29133...0.2..0.166.410.3j1......0....1..gws-wiz.......0i71j0i67j0i30.rEZiT1s1akM&ved=0ahUKEwj4w7qK2NLmAhVlzTgGHS8zA-MQ4dUDCAs&uact=5>

<https://github.com/jantimon/html-webpack-plugin>

<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>My Bioler plate app</title>

  </head>

  <body>

    <div id="app"></div>

  </body>

</html>

1. Now add plugins in webpack.config file. Which can be an array pass index.html inside.

module.exports = {

    entry:'./src/index.js',

    output:{

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

        filename: 'bundle.js'

    },

    module:{

        rules:[

            {

                test: /\.js$/,

                exclude: /node\_modules/,

                use:{

                    loader: 'babel-loader'

                }

            }

        ]

    },

    plugins:[

        new HtmlWebpackPlugin({

            template:'./src/index.html'

        })

    ]

}

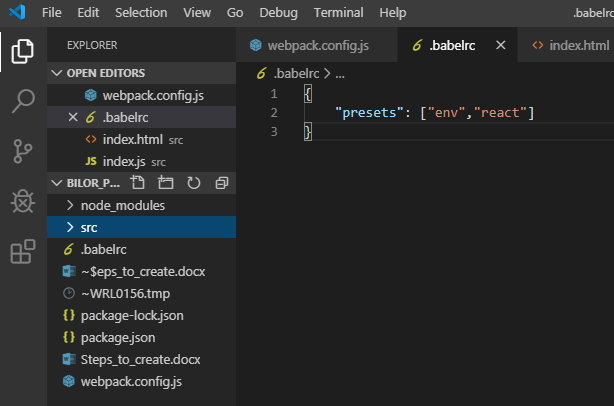
1. In order to use presets remember to install presets to compile ES6 and so on react presets we need to create a file **“.babelrc”** in the root directory.

This file has **JSON object** to include the presets. Here we have to include “presets” as an array, no need to include babel-preset like that just add [“env”,”react”].

{

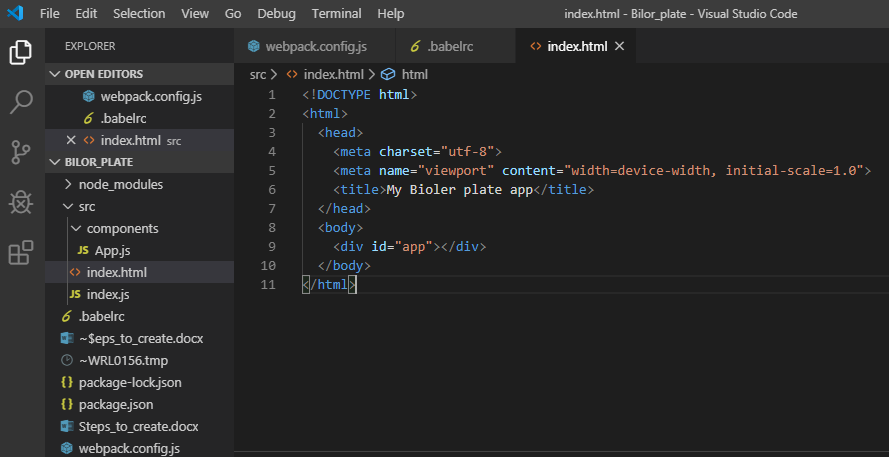
    "presets": ["env","react"]

}



**Step 7: React code**

Now in index.js add regular react code for Main dom. We are using babel so remember to write es6 syntax on it. So now create folder **“components”** inside the **“src”**, then create a file **“App.js”**



index.js:

import React from 'react';

import ReactDom from 'react-dom';

import App from './components/App';

ReactDom.render(<App/>,document.getElementById('app'));

App.js:

import React, { Component } from 'react';

class App extends Component{

    render(){

        return(

            <div>

                <h1>My first bolier plate app in React with Webpack</h1>

            </div>

        );

    }

}

export default App;

**Step 8: Add start script**

In package.json under **scripts{ }** add the key word **“start”** and place dev server **“webpack-dev-server”** options as follows

* **--mode development** - stands for development or production….
* **--open** - stands for open automatically when we run the command.
* --hot - stands for hot reloading when we save something it will reload the page

Now add the keyword **“build”** and place **“webpack”** it goanna use webpack cli to create dist folder. Options as follows

* **--mode production** - stands for development or production….

  "scripts": {

    "start": "webpack-dev-server --mode development --open --hot",

    "build": "webpack --mode production"

  },

**Step 9: start server**

Now type the command to start the dev server in your terminal.

> npm start

Working fine for this configurations

    "babel-core": "^6.26.0",

    "babel-loader": "^7.1.4",

    "babel-preset-env": "^1.6.1",

    "babel-preset-react": "^6.24.1",

    "html-webpack-plugin": "^3.1.0",

Once start the server we can get following messages in terminal

> bilor\_plate@1.0.0 start D:\React\Webpack\Bilor\_plate

> webpack-dev-server --mode development --open --hot

i ｢wds｣: Project is running at http://localhost:8080/

i ｢wds｣: webpack output is served from /

i ｢wds｣: Content not from webpack is served from D:\React\Webpack\Bilor\_plate

i ｢wdm｣: wait until bundle finished: /

i ｢wdm｣: Hash: 85adb09963f068095942

Version: webpack 4.41.4

Time: 8366ms

Built at: 2019-12-26 13:08:26

Asset Size Chunks Chunk Names

bundle.js 1.46 MiB main [emitted] main

index.html 300 bytes [emitted]

Entrypoint main = bundle.js

[0] multi (webpack)-dev-server/client?http://localhost:8080 (webpack)/hot/dev-server.js ./src/index.js 52 bytes {main} [built]

[./node\_modules/react-dom/index.js] 1.33 KiB {main} [built]

[./node\_modules/react/index.js] 190 bytes {main} [built]

[./node\_modules/webpack-dev-server/client/index.js?http://localhost:8080] (webpack)-dev-server/client?http://localhost:8080 4.29 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/overlay.js] (webpack)-dev-server/client/overlay.js 3.51 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/socket.js] (webpack)-dev-server/client/socket.js 1.53 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/utils/createSocketUrl.js] (webpack)-dev-server/client/utils/createSocketUrl.js 2.91 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/utils/log.js] (webpack)-dev-server/client/utils/log.js 964 bytes {main} [built]

[./node\_modules/webpack-dev-server/client/utils/reloadApp.js] (webpack)-dev-server/client/utils/reloadApp.js 1.59 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/utils/sendMessage.js] (webpack)-dev-server/client/utils/sendMessage.js 402 bytes {main} [built]

[./node\_modules/webpack-dev-server/node\_modules/strip-ansi/index.js] (webpack)-dev-server/node\_modules/strip-ansi/index.js 161 bytes {main} [built]

[./node\_modules/webpack/hot sync ^\.\/log$] (webpack)/hot sync nonrecursive ^\.\/log$ 170 bytes {main} [built]

[./node\_modules/webpack/hot/dev-server.js] (webpack)/hot/dev-server.js 1.59 KiB {main} [built]

[./node\_modules/webpack/hot/emitter.js] (webpack)/hot/emitter.js 75 bytes {main} [built]

[./src/index.js] 477 bytes {main} [built]

+ 32 hidden modules

Child html-webpack-plugin for "index.html":

1 asset

Entrypoint undefined = index.html

[./node\_modules/html-webpack-plugin/lib/loader.js!./src/index.html] 458 bytes {0} [built]

[./node\_modules/lodash/lodash.js] 528 KiB {0} [built]

[./node\_modules/webpack/buildin/global.js] (webpack)/buildin/global.js 472 bytes {0} [built]

[./node\_modules/webpack/buildin/module.js] (webpack)/buildin/module.js 497 bytes {0} [built]

i ｢wdm｣: Compiled successfully.

But this configuration getting error

    "babel-core": "^6.26.3",

    "babel-loader": "^8.0.6",

    "babel-preset-env": "^1.7.0",

    "babel-preset-react": "^6.24.1",

    "html-webpack-plugin": "^3.2.0",

> bilor\_plate@1.0.0 start D:\React\Webpack\Bilor\_plate

> webpack-dev-server --mode development --open --hot

i ｢wds｣: Project is running at http://localhost:8080/

i ｢wds｣: webpack output is served from /

i ｢wds｣: Content not from webpack is served from D:\React\Webpack\Bilor\_plate

i ｢wdm｣: wait until bundle finished: /

× ｢wdm｣: Hash: 58acda0bdd9b39c1b974

Version: webpack 4.41.4

Time: 1258ms

Built at: 2019-12-26 13:12:16

Asset Size Chunks Chunk Names

bundle.js 391 KiB main [emitted] main

index.html 300 bytes [emitted]

Entrypoint main = bundle.js

[0] multi (webpack)-dev-server/client?http://localhost:8080 (webpack)/hot/dev-server.js ./src/index.js 52 bytes {main} [built]

[./node\_modules/webpack-dev-server/client/index.js?http://localhost:8080] (webpack)-dev-server/client?http://localhost:8080 4.29 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/overlay.js] (webpack)-dev-server/client/overlay.js 3.51 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/socket.js] (webpack)-dev-server/client/socket.js 1.53 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/utils/createSocketUrl.js] (webpack)-dev-server/client/utils/createSocketUrl.js 2.91 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/utils/log.js] (webpack)-dev-server/client/utils/log.js 964 bytes {main} [built]

[./node\_modules/webpack-dev-server/client/utils/reloadApp.js] (webpack)-dev-server/client/utils/reloadApp.js 1.59 KiB {main} [built]

[./node\_modules/webpack-dev-server/client/utils/sendMessage.js] (webpack)-dev-server/client/utils/sendMessage.js 402 bytes {main} [built]

[./node\_modules/webpack-dev-server/node\_modules/strip-ansi/index.js] (webpack)-dev-server/node\_modules/strip-ansi/index.js 161 bytes {main} [built]

[./node\_modules/webpack/hot sync ^\.\/log$] (webpack)/hot sync nonrecursive ^\.\/log$ 170 bytes {main} [built]

[./node\_modules/webpack/hot/dev-server.js] (webpack)/hot/dev-server.js 1.59 KiB {main} [built]

[./node\_modules/webpack/hot/emitter.js] (webpack)/hot/emitter.js 75 bytes {main} [built]

[./node\_modules/webpack/hot/log-apply-result.js] (webpack)/hot/log-apply-result.js 1.27 KiB {main} [built]

[./node\_modules/webpack/hot/log.js] (webpack)/hot/log.js 1.34 KiB {main} [built]

[./src/index.js] 810 bytes {main} [not cacheable] [built] [failed] [1 error]

+ 20 hidden modules

ERROR in ./src/index.js

Module build failed (from ./node\_modules/babel-loader/lib/index.js):

Error: Cannot find module '@babel/core'

babel-loader@8 requires Babel 7.x (the package '@babel/core'). If you'd like to use Babel 6.x ('babel-core'), you should install 'babel-loader@7'.

at Function.Module.\_resolveFilename (module.js:547:15)

at Function.Module.\_load (module.js:474:25)

at Module.require (module.js:596:17)

at require (internal/module.js:11:18)

at Object.<anonymous> (D:\React\Webpack\Bilor\_plate\node\_modules\babel-loader\lib\index.js:10:11)

at Module.\_compile (module.js:652:30)

at Object.Module.\_extensions..js (module.js:663:10)

at Module.load (module.js:565:32)

at tryModuleLoad (module.js:505:12)

at Function.Module.\_load (module.js:497:3)

Child html-webpack-plugin for "index.html":

1 asset

Entrypoint undefined = index.html

[./node\_modules/html-webpack-plugin/lib/loader.js!./src/index.html] 458 bytes {0} [built]

[./node\_modules/lodash/lodash.js] 528 KiB {0} [built]

[./node\_modules/webpack/buildin/global.js] (webpack)/buildin/global.js 472 bytes {0} [built]

[./node\_modules/webpack/buildin/module.js] (webpack)/buildin/module.js 497 bytes {0} [built]

i ｢wdm｣: Failed to compile.

**Solutions:**

**Upgrade babel as mentioned below**

babel-loader@8 uses  Babel 7.x and this requires @babel/core instead of babel-core. Sounds like you either accidentally upgraded babel-loader or forgot to install @babel/core. So need to install @babel/core instead of babel-core and same for @babel/preset-env => babel-preset-env then @babel/preset-react => babel-preset-react

<https://github.com/babel/gulp-babel/issues/124>

<https://github.com/babel/babel-loader/issues/664>

**Install**

<https://www.npmjs.com/package/@babel/core>

<https://www.npmjs.com/package/@babel/cli>

<https://www.npmjs.com/package/@babel/preset-env>

<https://www.npmjs.com/package/@babel/preset-react>

**Update .babelrc file**

Just like env is now @babel/env, react should be @babel/react and you'll need to install @babel/preset-react. So place this code

{

    "presets": ["@babel/env", "@babel/react"]

}

Instead of

{

    "presets": ["env", "react"]

}

Ref:

<https://github.com/babel/babel/issues/6808>