

برپا کردن محیط ReactJS

در این بخش می خواهیم به چگونگی برپا کردن محیط جهت برنامه نویسی موفق در ری اکت بپردازیم. توجه داشته باشید که برای انجام این کار مراحل زیادی نیاز است، اما این کارها در آینده فرآیند برنامه نویسی را سرعت می بخشند. برای انجام این کار به NodeJS نیاز خواهیم داشت. بنابراین اگر آن را نصب نکرده اید، به لینک جدول زیر مراجعه کنید.

ردیف	نرم افزار و توضیحات
1	NPM و NodeJS NodeJS پلتفرمی است که برای توسعه ی ReactJS به آن نیاز است. به برپا کردن محیط NodeJS مراجعه کنید.

پس از نصب موفق NodeJS می توانیم با استفاده از npm ، React را نصب کنیم. به دو صورت زیر می توانید NodeJS را نصب کنید:

- استفاده از webpack و babel.
- استفاده از دستور create-react-app

نصب ReactJS با استفاده از webpack و babel

وب پک یک ماژول مجموعه ای است (ماژول های مستقل را مدیریت کرده و آن ها را بارگیری می کند). وب پک ماژول های وابسته را گرفته و آن ها را به یک فایل واحد کامپایل می کند. طی برنامه نویسی با استفاده از خط فرمان یا با پیکربندی آن با استفاده از فایل webpack.config می توانید از وب پک استفاده کنید.

بابل یک کامپایلر و ترنسپایلر جاوا اسکریپت است. Babel در تبدیل یک سورس کد به سورس کد دیگر کاربرد دارد. با استفاده از این قابلیت می توانید در کد خود از امکانات جدید ES6 استفاده کنید، به گونه ای که بابل این امکانات را به ES5 قدیمی و ساده تبدیل می کند تا بتوان از آن ها در تمامی مرورگرها استفاده کرد.

مرحله 1 : ایجاد پوشه ی اصلی

در دسکتاپ پوشه ای با نام reactApp ایجاد کنید تا بتوانید با استفاده از دستور mkdir فایل های مورد نیاز را نصب کنید.

```
C:\Users\username\Desktop>mkdir reactApp
C:\Users\username\Desktop>cd reactApp
```

برای این که بتوانید هر ماژولی را ایجاد کنید، نیاز است که فایل package.json را تولید کنید. بنابراین پس از ایجاد این پوشه ما باید یک فایل package.json را ایجاد کنیم. برای انجام این کار از دستور npm init وارد کنید.

```
C:\Users\username\Desktop\reactApp>npm init
```

این دستور اطلاعاتی در رابطه با ماژول را درخواست می کند؛ اطلاعاتی مانند اسم بسته، توضیحات، ناشر و ... با استفاده از گزینه ی -y می توانید از این مرحله عبور کنید.

```
C:\Users\username\Desktop\reactApp>npm init -y
```

Wrote to C:\reactApp\package.json:

```
{
  "name": "reactApp",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

مرحله 2: نصب ری اکت و react dom

با توجه به این که کار اصلی ما نصب ReactJS است، جهت نصب آن و بسته های DOM آن به ترتیب از دستورات `install react` و `react-dom` متعلق به `npm` استفاده کنید. می توانید با گزینه `--save` بسته هایی که ما نصب کرده ایم را به فایل `package.json` اضافه کنید.

```
C:\Users\Tutorialspoint\Desktop\reactApp>npm install react --save
C:\Users\Tutorialspoint\Desktop\reactApp>npm install react-dom --save
```

یا می توانید تمامی آن ها را مانند زیر در یک دستور واحد نصب کنید.

```
C:\Users\username\Desktop\reactApp>npm install react react-dom --save
```

مرحله 3: نصب وب پک

با توجه به این که ما جهت تولید این مجموعه در حال استفاده از وب پک هستیم، `webpack-dev-server` و `webpack-cli` را نصب کنید.

```
C:\Users\username\Desktop\reactApp>npm install webpack --save
C:\Users\username\Desktop\reactApp>npm install webpack-dev-server --save
C:\Users\username\Desktop\reactApp>npm install webpack-cli --save
```

یا می توانید تمامی آن ها را مانند زیر در یک دستور واحد نصب کنید.

```
C:\Users\username\Desktop\reactApp>npm install webpack webpack-dev-server webpack-cli --save
```

مرحله 4: نصب Babel

بابل و پلاگین های `babel-core`، `babel-loader`، `babel-preset-env`، `babel-preset-react` و `html-` `webpack-plugin` را نصب کنید.

```
C:\Users\username\Desktop\reactApp>npm install babel-core --save-dev
C:\Users\username\Desktop\reactApp>npm install babel-loader --save-dev
C:\Users\username\Desktop\reactApp>npm install babel-preset-env --save-dev
C:\Users\username\Desktop\reactApp>npm install babel-preset-react --save-dev
C:\Users\username\Desktop\reactApp>npm install html-webpack-plugin --save-dev
```

یا می توانید مانند زیر تمامی آن ها را در یک دستور واحد نصب کنید.

```
C:\Users\username\Desktop\reactApp>npm install babel-core babel-loader babel-preset-env
```

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

مرحله 5: ایجاد فایل ها

جهت تکمیل نصب باید برخی از فایل های خاص را مانند `index.html`، `App.js`، `main.js`، `webpack.config.js` و `babelrc` ایجاد کنیم. می توانید این فایل ها را یا به صورت دستی و یا با استفاده از `cmd` ایجاد کنید.

```
C:\Users\username\Desktop\reactApp>type nul > index.html
C:\Users\username\Desktop\reactApp>type nul > App.js
C:\Users\username\Desktop\reactApp>type nul > main.js
C:\Users\username\Desktop\reactApp>type nul > webpack.config.js
C:\Users\username\Desktop\reactApp>type nul > .babelrc
```

مرحله 6: تنظیم کامپایلر، سرور و لودرها

فایل `webpack.config.js` را باز کنید و کد زیر را به آن اضافه کنید. می خواهیم نقطه ی ورودی وب پک را به گونه ای تنظیم کنیم که این نقطه `main.js` شود. مسیر خروجی جایی است که سرور به برنامه خدمات می رساند. همچنین سرور برنامه نویسی را بر روی پورت 8001 تنظیم می کنیم. این پورت را می توانید به صورت دلخواه نیز انتخاب کنید.

`webpack.config.js`

```
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'

  })

]

}
```

package.json را باز کنید و "test" "echo \"Error: no test specified\" && exit 1" را از داخل شیء "scripts" پاک کنید. دلیل پاک کردن این خط این است که ما در این آموزش نیازی به آزمایش نخواهیم داشت. بیا بعد در عوض دستورات start و build را اضافه کنیم.

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

مرحله 7: index.html

این مرحله صرفاً یک HTML معمولی است. "div id = \"app\" را به عنوان عنصر اصلی برنامه تنظیم می‌کنیم و اسکریپت index_bundle.js که در واقع فایل برنامه‌ی ما است را اضافه می‌کنیم.

```
<!DOCTYPE html>  
  
<html lang = \"en\">  
  
  <head>  
  
    <meta charset = \"UTF-8\">  
  
    <title>React App</title>  
  
  </head>  
  
  <body>  
  
    <div id = \"app\"></div>  
  
    <script src = 'index_bundle.js'></script>  
  
  </body>  
  
</html>
```

مرحله 8: App.jsx و main.js

این مرحله اولین جزء ری‌اکت را دربر می‌گیرد. در یکی از بخش‌های بعدی به صورت مفصل به اجزای ری‌اکت خواهیم پرداخت. این جزء باعث می‌شود Hello World رندر شود.

App.js

```
import React, { Component } from 'react';

class App extends Component{

  render(){

    return(

      <div>

        <h1>Hello World</h1>

      </div>

    );

  }

}

export default App;
```

باید این جزء را ایمپورت کنیم و آن را در عنصر اصلی برنامه ی خود رندر کنیم تا بتوانیم آن را در مرورگر ببینیم.

main.js

```
import React from 'react';

import ReactDOM from 'react-dom';

import App from './App.js';

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

نکته: هر زمان که بخواهید از چیزی استفاده کنید، باید آن را در مرحله ی اول ایمپورت کنید. اگر می خواهید این جزء در بخش های دیگر برنامه قابل استفاده باشد، باید آن را پس از ایجاد export کنید و آن را در فایل که می خواهید از آن استفاده کنید، import کنید.

فایلی را با نام babelrc ایجاد کنید و محتوای زیر را در آن کپی کنید.

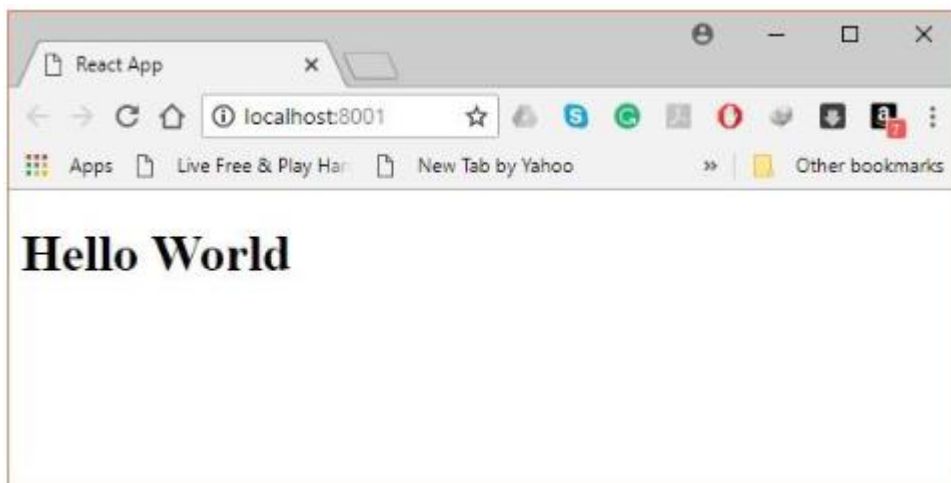
```
{
  "presets":["env", "react"]
}
```

مرحله 9: اجرای سرور

عملیات برپا کردن محیط تمام شده است و می توانیم با اجرای دستور زیر سرور را راه اندازی کنیم.

```
C:\Users\username\Desktop\reactApp>npm start
```

با اجرای این کار پورتی که ما می خواهیم در مرورگر باز شود، نمایش داده می شود. این پورت برای ما <http://localhost:8001/> است. بعد از این که این پورت را باز کنیم، نتیجه ی زیر نمایش داده می شود.

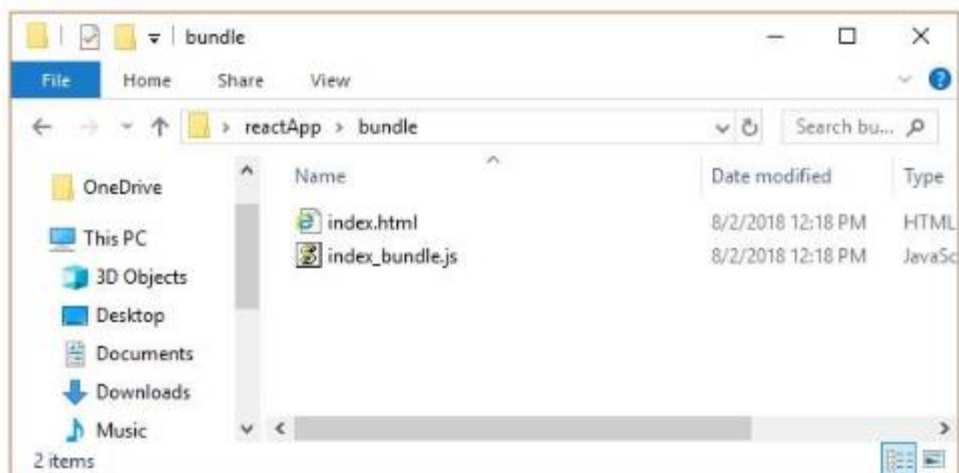


مرحله 10: تولید نرم افزار

در نهایت برای آن که بتوانید نرم افزار را تولید کنید، باید دستور `build` را در `cmd` مانند زیر اجرا کنید.

```
C:\Users\Tutorialspoint\Desktop\reactApp>npm run build
```

این کار باعث می شود این نرم افزار مانند زیر در پوشه ی فعلی تولید شود.



استفاده از دستور create-react-app

به جای استفاده از وب پک و بابل می توانید به صورت ساده تری ReactJS را با نصب create-react-app نصب کنید.

مرحله 1: نصب create-react-app

به دسکتاپ بروید و با استفاده از cmd مانند زیر Create React App نصب کنید.

```
C:\Users\Tutorialspoint>cd C:\Users\Tutorialspoint\Desktop\  
C:\Users\Tutorialspoint\Desktop>npx create-react-app my-app
```

این کار باعث می شود، در دسکتاپ پوشه ای به نام my-app ایجاد شود و تمامی فایل های مورد نیاز در آن نصب شود.

مرحله 2: حذف تمامی فایل های منبع

به پوشه ی SRC موجود در پوشه ی my-app تولید شده بروید و مانند زیر تمامی فایل ها را حذف کنید.

```
C:\Users\Tutorialspoint\Desktop>cd my-app/src  
C:\Users\Tutorialspoint\Desktop\my-app\src>del *  
C:\Users\Tutorialspoint\Desktop\my-app\src*, Are you sure (Y/N)? y
```

مرحله 3: اضافه کردن فایل ها

با استفاده از نام های `index.css` و `index.js` موجود در پوشه `src` فایل ها را اضافه کنید.

```
C:\Users\Tutorialspoint\Desktop\my-app\src>type nul > index.css  
C:\Users\Tutorialspoint\Desktop\my-app\src>type nul > index.js
```

کد زیر را به فایل `index.js` اضافه کنید.

```
import React from 'react';  
import ReactDOM from 'react-dom';  
import './index.css';
```

مرحله 4: اجرای پروژه

در نهایت با استفاده از دستور `start` پروژه را اجرا کنید.

```
npm start
```

