**通过npm使用React**

2016年12月5日

10:25

要想使用npm,请先安装nodejs.

第一步、安装全局包

$ npm install babel -g

$ npm install webpack -g

$ npm install webpack-dev-server -g

第二步、创建根目录

创建一个根目录，目录名为：reactApp，再使用 npm init 初始化，生成 package.json 文件：

$ mkdir reactApp

$ cd reactApp/

$ npm init

name: (reactApp) runoob-react-test

version: (1.0.0)

description: cllgeek

entry point: (index.js)

test command:

git repository:

keywords:

author:

license: (ISC)

About to write to /Users/tianqixin/www/reactApp/package.json:

{

"name": "react-runoob",

"version": "1.0.0",

"description": "cllgeek test",

"main": "index.js",

"scripts": {

"start": "webpack-dev-server --hot"

},

"author": "",

"license": "ISC",

"dependencies": {

"react": "^0.14.8",

"react-dom": "^0.14.8"

}

}

Is this ok? (yes)

第三步、添加越来包及插件

因为我们要使用 React, 所以我们需要先安装它，–save 命令用于将包添加至 package.json 文件。

$ npm install react --save

$ npm install react-dom --save

同时我们也要安装一些 babel 插件

$ npm install babel-core //babel主体

$ npm install babel-loader //babel动态编译？

$ npm install babel-preset-react //插件包，

$ npm install babel-preset-es2015 //插件包，支持es6语法等

第四步、创建文件

接下来我们创建一些必要文件：

$ touch index.html

$ touch App.jsx

$ touch main.js

$ touch webpack.config.js

第五步、设置编译器，服务器，载入器

打开 webpack.config.js 文件添加以下代码:

var config = {

entry: './main.js',

output: {

path:'./',

filename: 'index.js',

},

devServer: {

inline: true,

port: 7777

},

module: {

loaders: [ {

test: /.jsx?$/,

exclude: /node\_modules/,

loader: 'babel',

query: {  
 presets: ['es2015', 'react']  
 }  
 }]

}

}

module.exports = config;

entry: 指定打包的入口文件 main.js。

output：配置打包结果，path定义了输出的文件夹，filename则定义了打包结果文件的名称。

devServer：设置服务器端口号为 7777，端口后你可以自己设定 。

module：定义了对模块的处理逻辑，这里可以用loaders定义了一系列的加载器，以及一些正则。当需要加载的文件匹配test的正则时，就会调用后面的loader对文件进行处理，这正是webpack强大的原因。

现在打开 package.json 文件，找到 “scripts” 中的 “test” “echo \”Error: no test specified\” && exit 1″ 使用以下代码替换：

"start": "webpack-dev-server --hot"

替换后的 package.json 文件 内容如下：

$ cat package.json

{

"name": "runoob-react-test",

"version": "1.0.0",

"description": "cllgeek test",

"main": "index.js",

"scripts": {

"start": "webpack-dev-server --hot"

},

"author": "",

"license": "ISC",

"dependencies": {

"react": "^0.14.7",

"react-dom": "^0.14.7"

}

}

现在我们可以使用 npm start 命令来启动服务。–hot 命令会在文件变化后重新载入，这样我们就不需要在代码修改后重新刷新浏览器就能看到变化。

第六步、index.html

设置 div id = “app” 为我们应用的根元素，并引入 index.js 脚本文件。

<!DOCTYPE html>

<html>

<head>

<meta charset = "UTF-8">

<title>React App - cllgeek(cllgeek.com)</title>

</head>

<body>

<div id = "app"></div>

<script src = "index.js"></script>

</body>

</html>

第七步、App.jsx 和 main.js

这是第一个 react 组件。后面的章节我们会详细介绍 React 组件。这个组件将输出 Hello World!!!。

App.jsx 文件代码

import React from 'react';

class App extends React.Component {

render() {

return (

<div>

Hello World!!!<br />

cll极客网

</div>

);

}

}

export default App;

我们需要引入组件并将其渲染到根元素 App 上，这样我们才可以在浏览器上看到它。

main.js 文件代码

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App.jsx';

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

注意：

如果想要组件可以在任何的应用中使用，需要在创建后使用 export 将其导出，在使用组件的文件使用 import将其导入。

第八步、运行服务

完成以上配置后，我们即可运行该服务：

$ npm start

通过浏览器访问 [http://localhost:7777/，输出结果如下](http://localhost:7777/%EF%BC%8C%E8%BE%93%E5%87%BA%E7%BB%93%E6%9E%9C%E5%A6%82%E4%B8%8B):

QQ图片20160405094825

作者： cllgeek

链接：<http://www.imooc.com/article/6219>

来源：慕课网

源文档 <<http://www.imooc.com/article/6219>>