React使用create-react-app快速搭建工程

# 创建工程

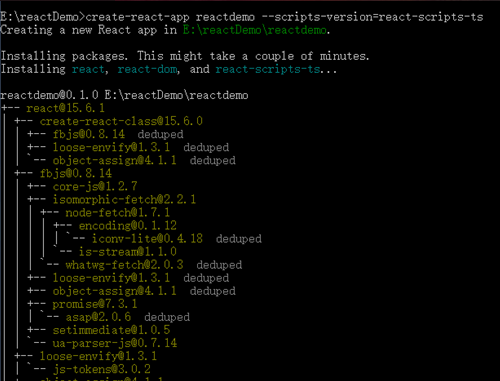
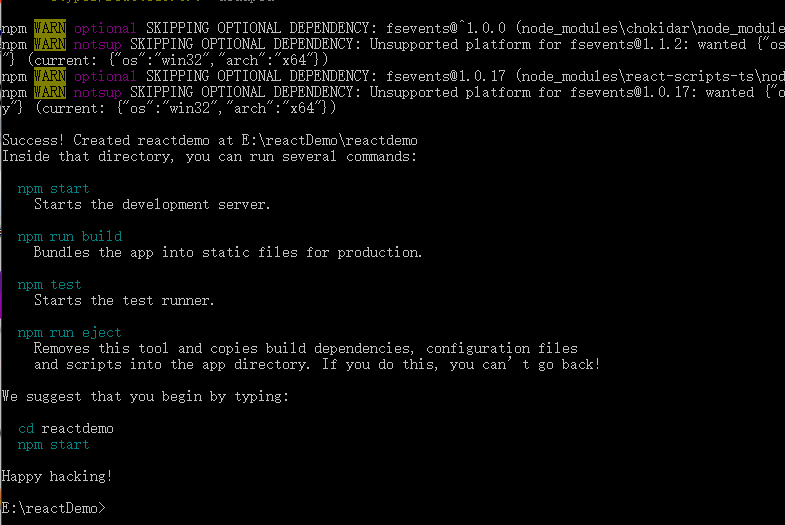
## 安装create-react-app

npm安装命令：npm install -g create-react-app

## 创建React-typescript工程

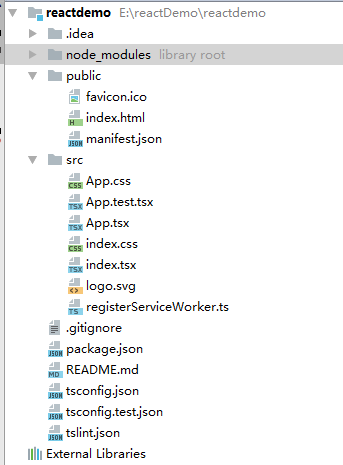
create-react-app project\_name --scripts-version=react-scripts-ts

注意：project\_name是你的项目名称；

上图就是创建过程，创建过程中，npm需要下载相关组件，请耐心等待。

工程结构图如下



这样，最简单的react-typescript工程就创建好了。

工程结构：

node\_modules:npm依赖组件

public：静态资源

src：项目主要工程目录

package.json：npm配置文件

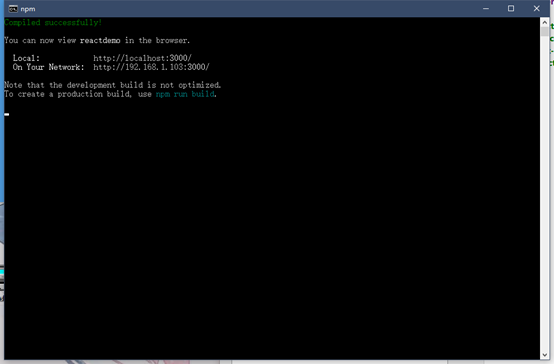
tsconfig.json：typescript的配置文件

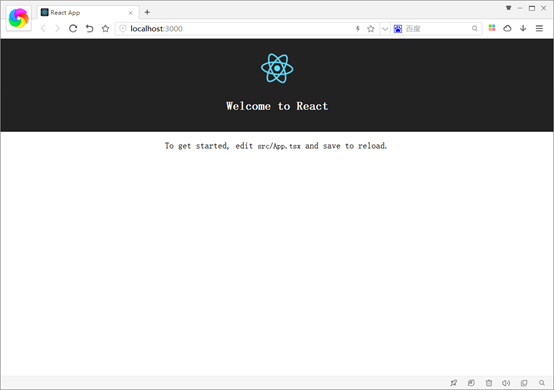
tslint.json：typescript的语法规范配置文件

## 启动

打开工程目录命令：cd project\_name

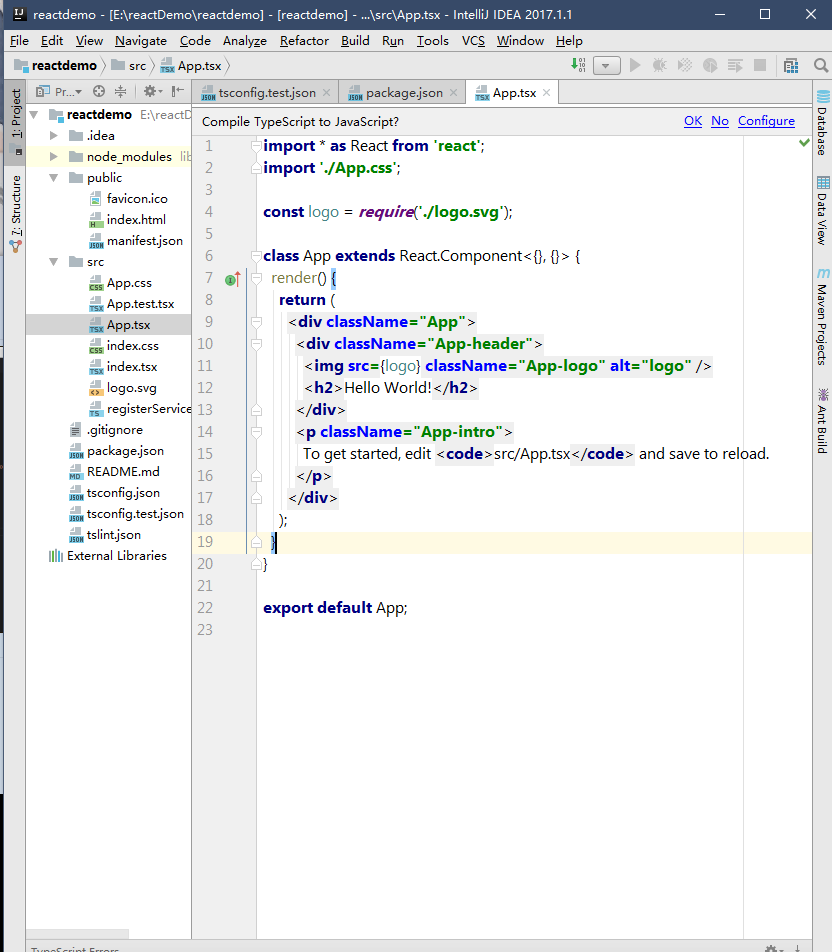
启动命令：npm start

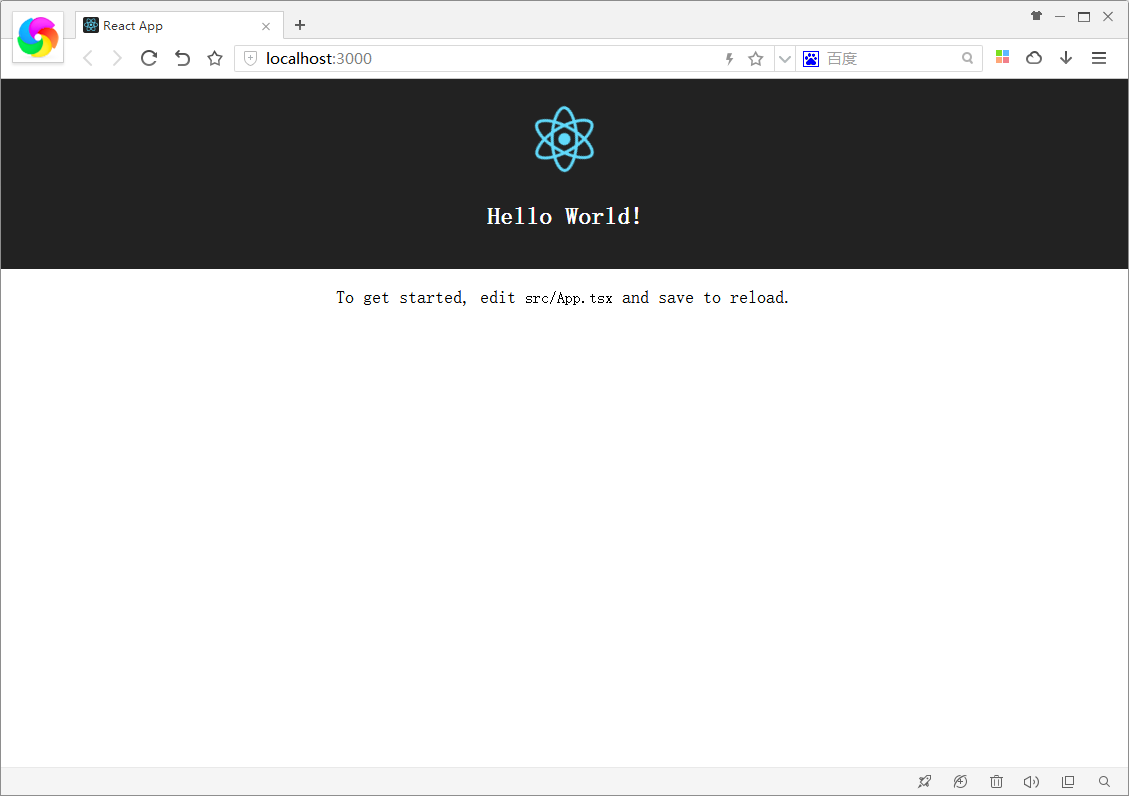


浏览器输入http://localhost:3000/就可以看到ReactApp啦

## 测试

可以尝试将App.tsx中的Welcome to React修改为Hello World!

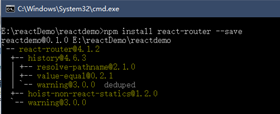


热部署已生效：

# 整合路由-Router-前后端分离

## 安装react-router

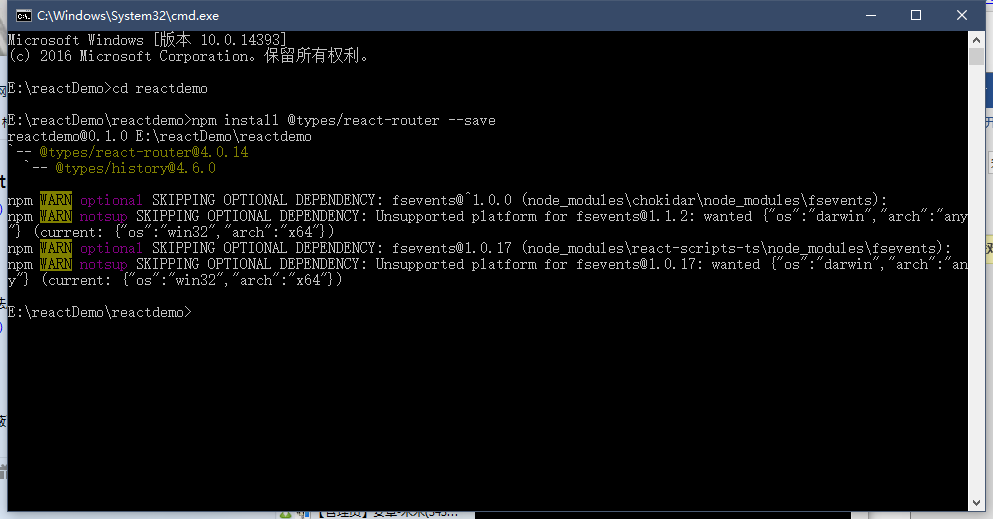
npm安装命令：npm install --save react-router



安装typescript的react-router index.d.ts

npm安装命令：*npm install @types/react-router –save*

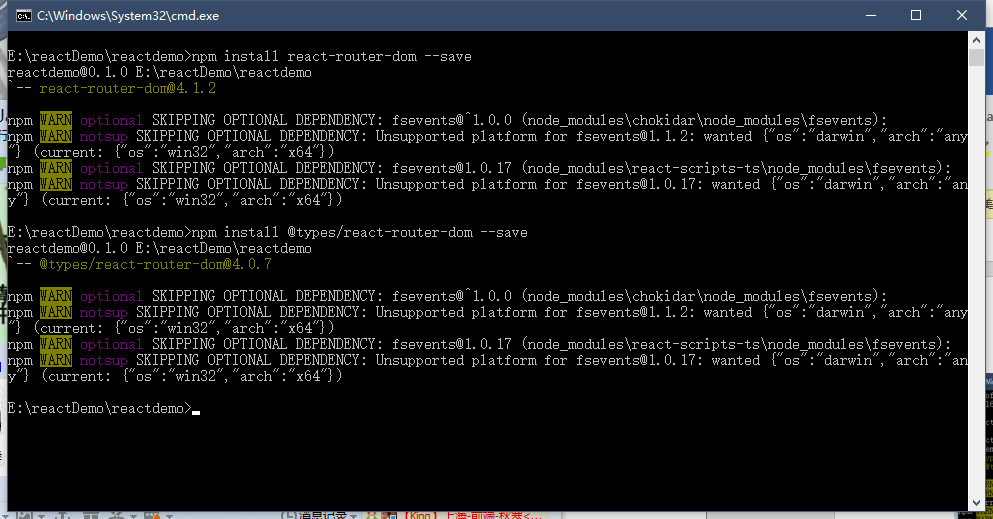
注意：工程使用的typescript所以需要安装@types/react-router；



安装react-router-dom及typescript的index.d.tsx：

npm安装命令：*npm install react-router-dom –save*

npm安装命令：*npm install @types/react-router-dom –save*

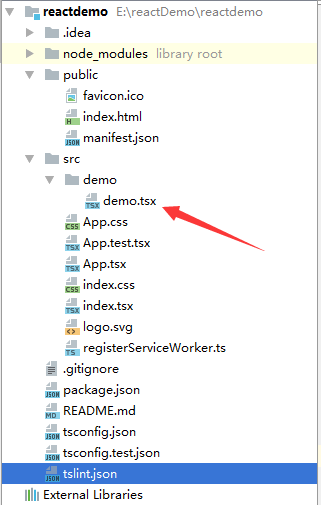


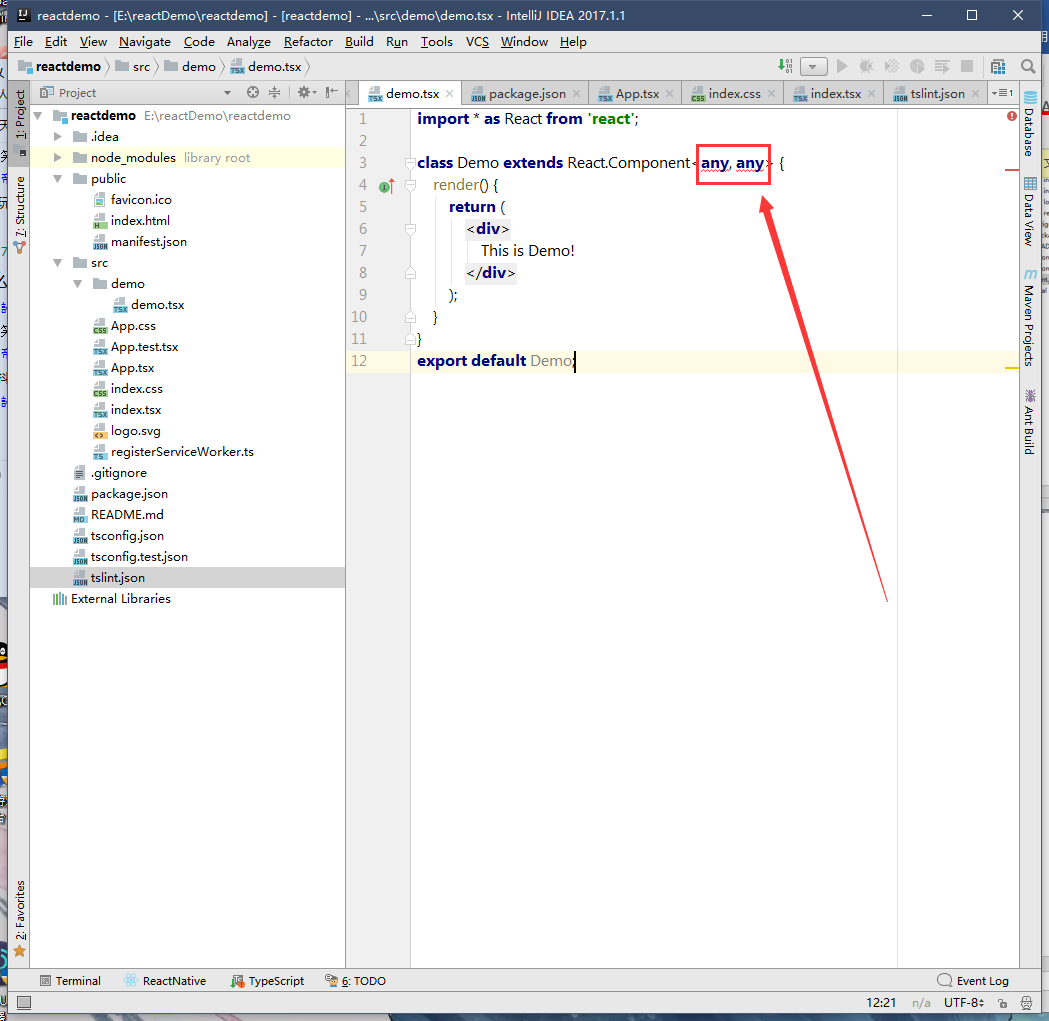
## 使用

### 1.创建demo组件

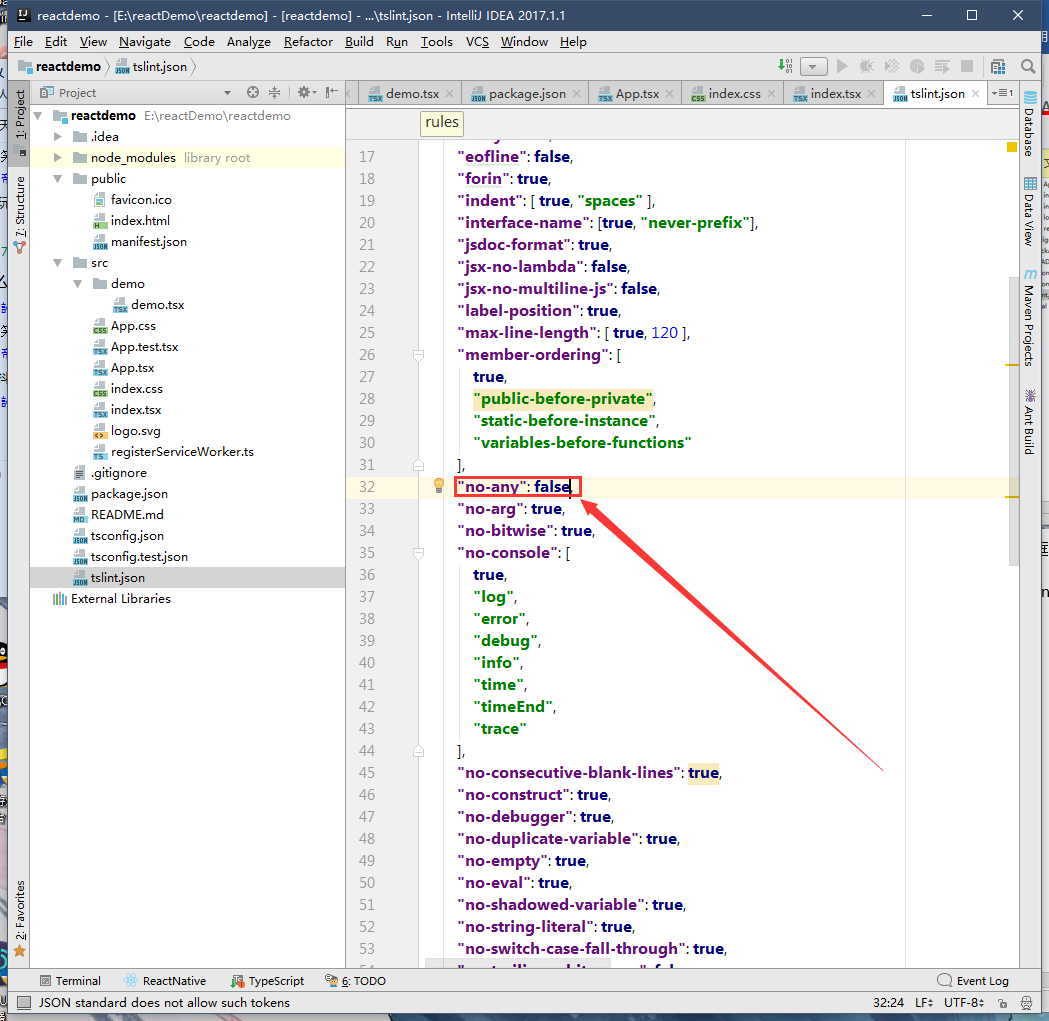
在src目录下创建demo目录：

创建demo.tsx

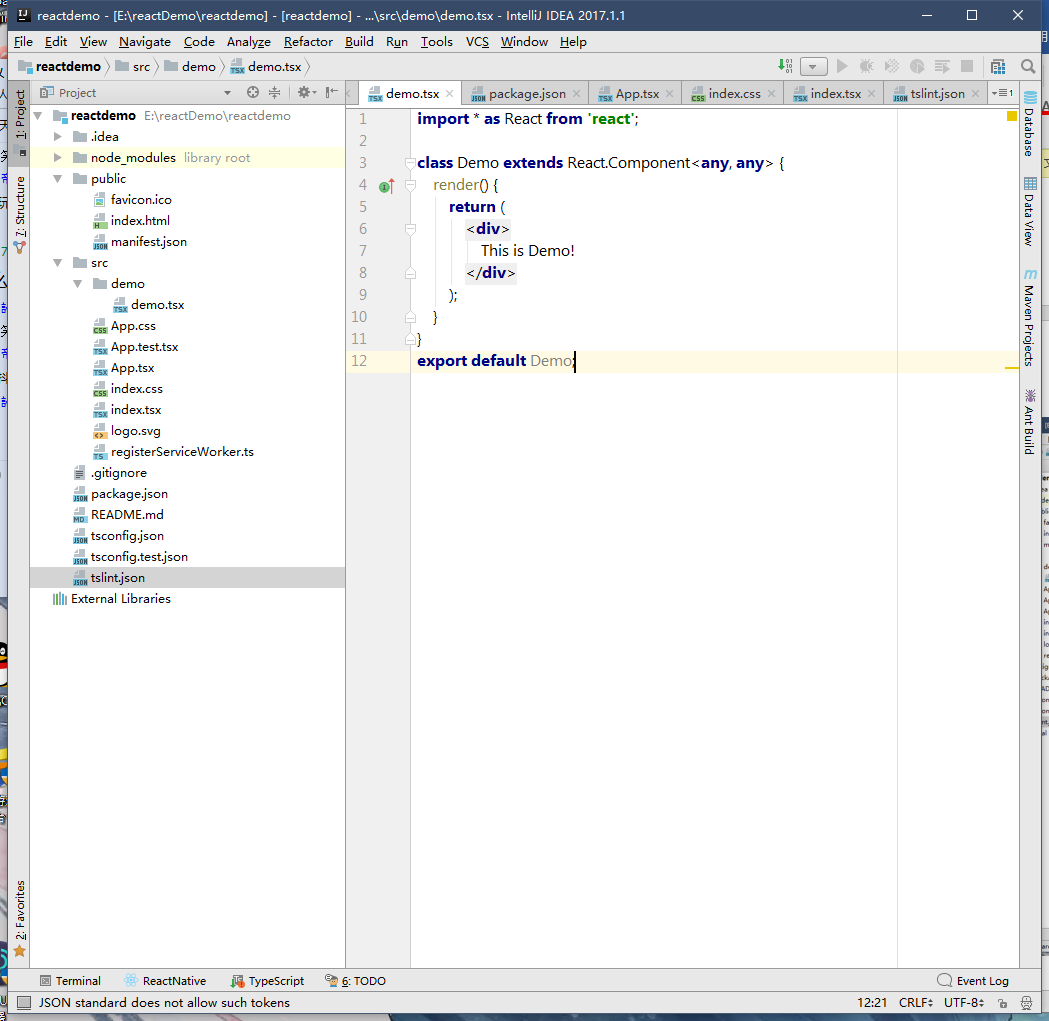


在demo.tsx中输入：

注意：其中红框中会报语法错误，这是因为tslint中的any配置为true。请将tslint.json中的no-any置为false:

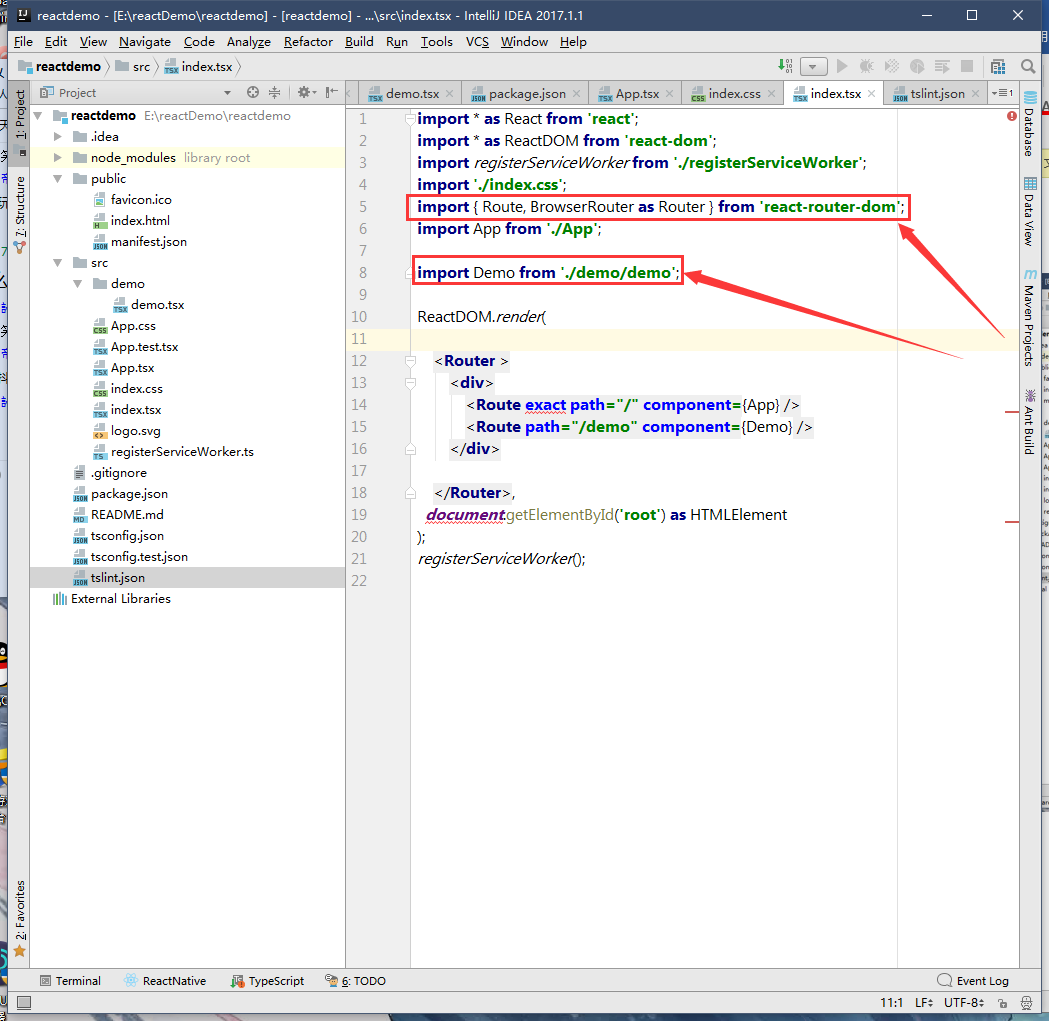


这样，tslint就不会报语法错误了。



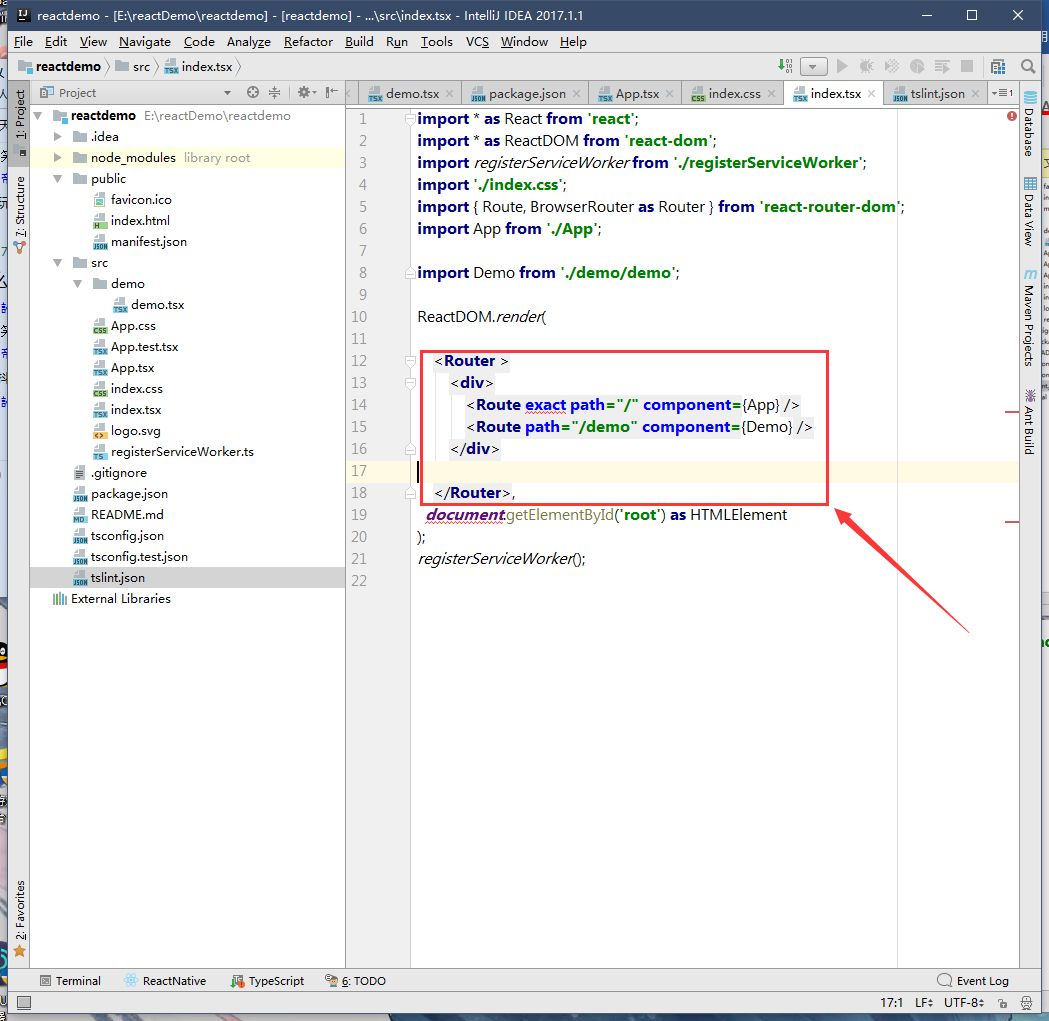
### 2.配置路由

在index.tsx中配置路由



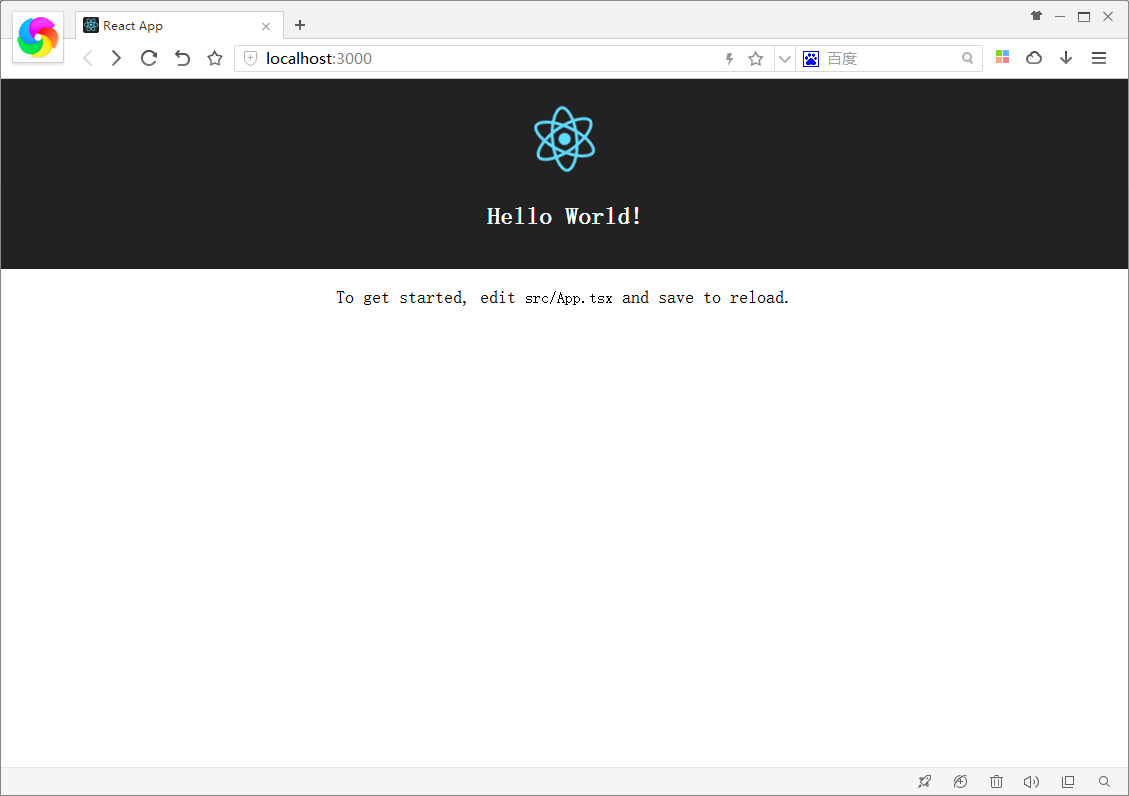
引入react-router-dom到index.tsx中

然后配置路由，如下图

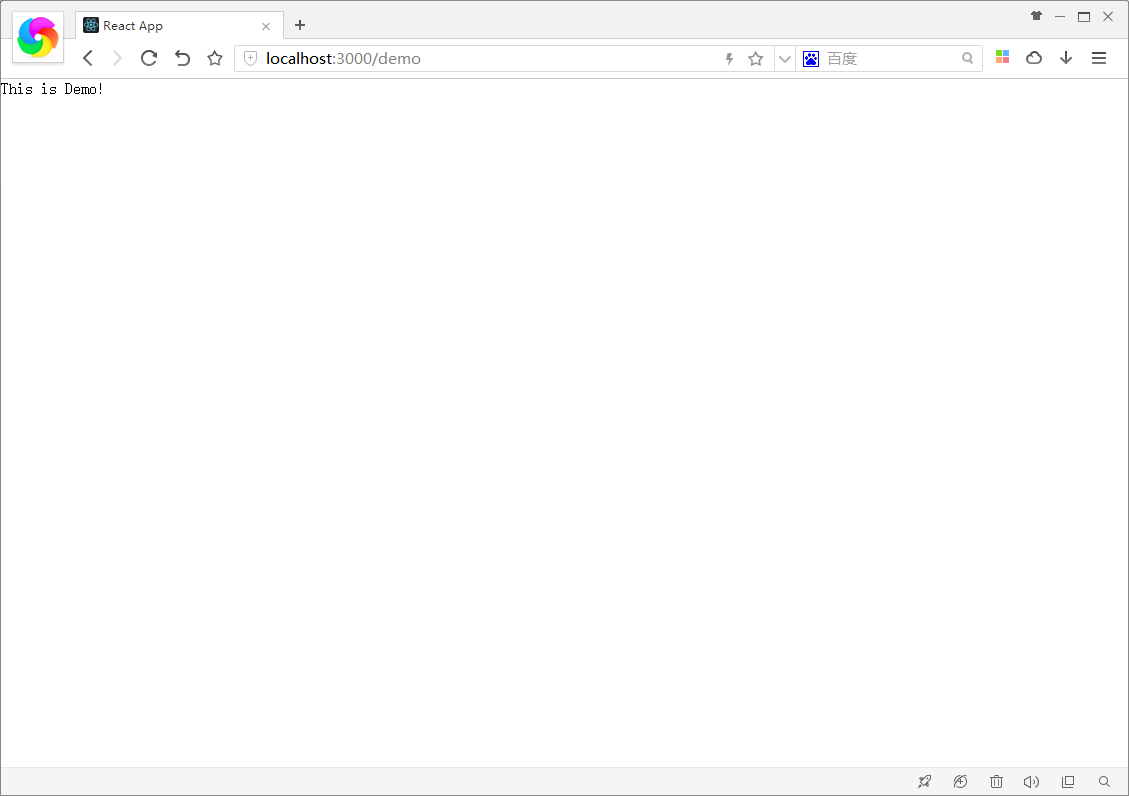


## 测试

切换到浏览器，输入地址：http://localhost:3000/

在默认根路径下，就展示App组件

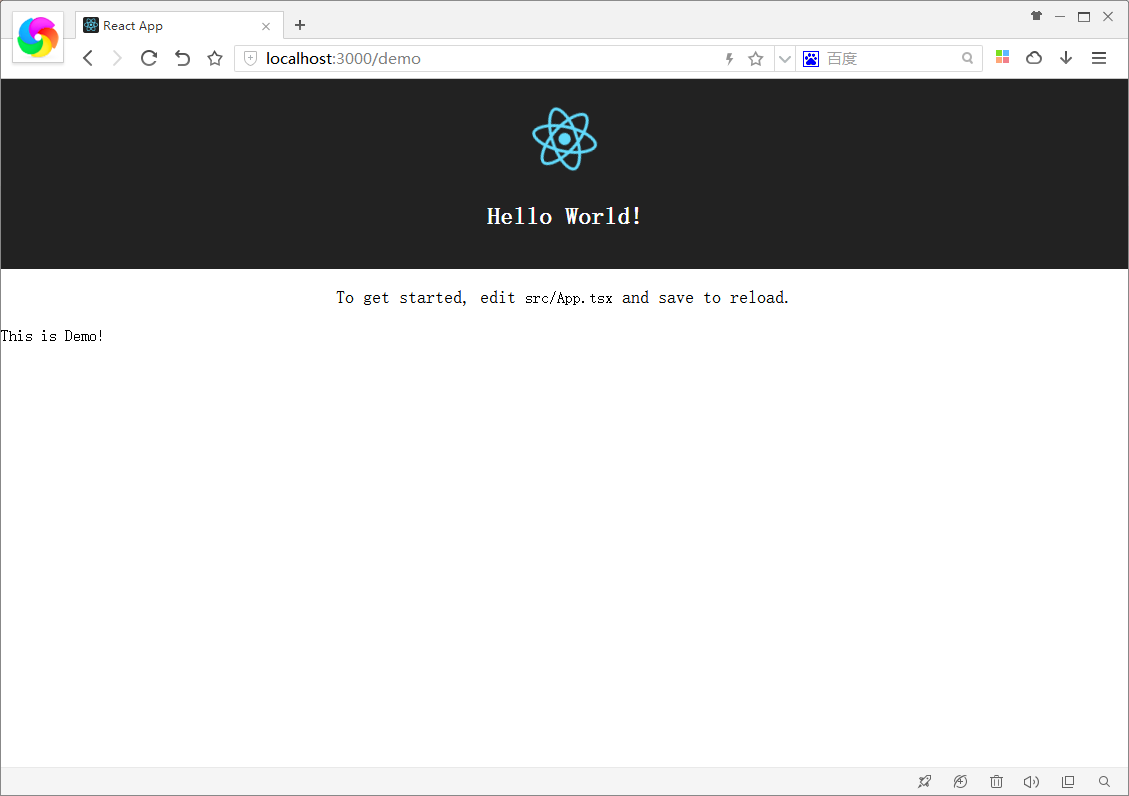
然后在浏览器输入<http://localhost:3000/demo>，这时就会展示demo组件中的内容：



注意：

<Route exact path="/" component={App} />

中的exact属性是指路径精确匹配，也就是如果不加这个属性，当切换到Demo页面时也会出现App组件的内容,如图：



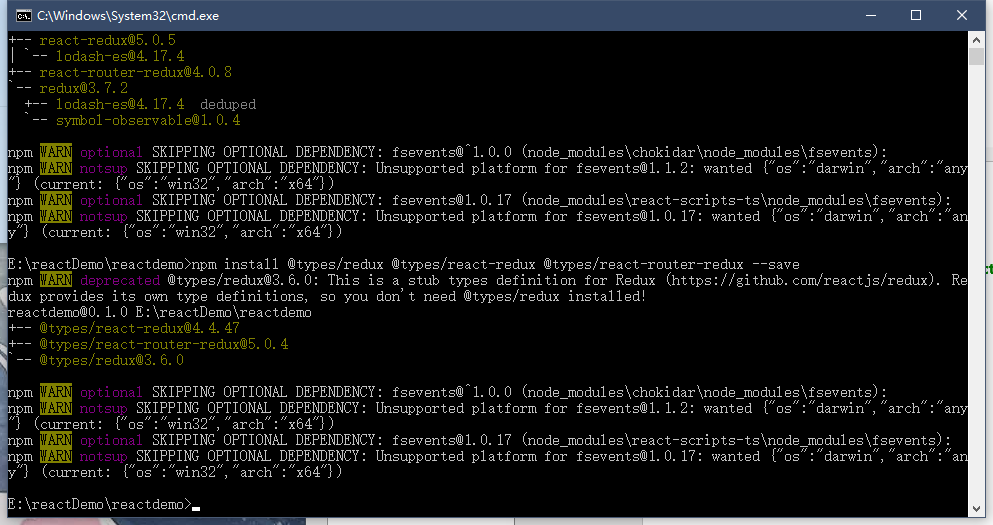
# 整合redux-实现组件间的通信

## 安装

安装redux，react-redux，react-router-redux，包括他们的typescript

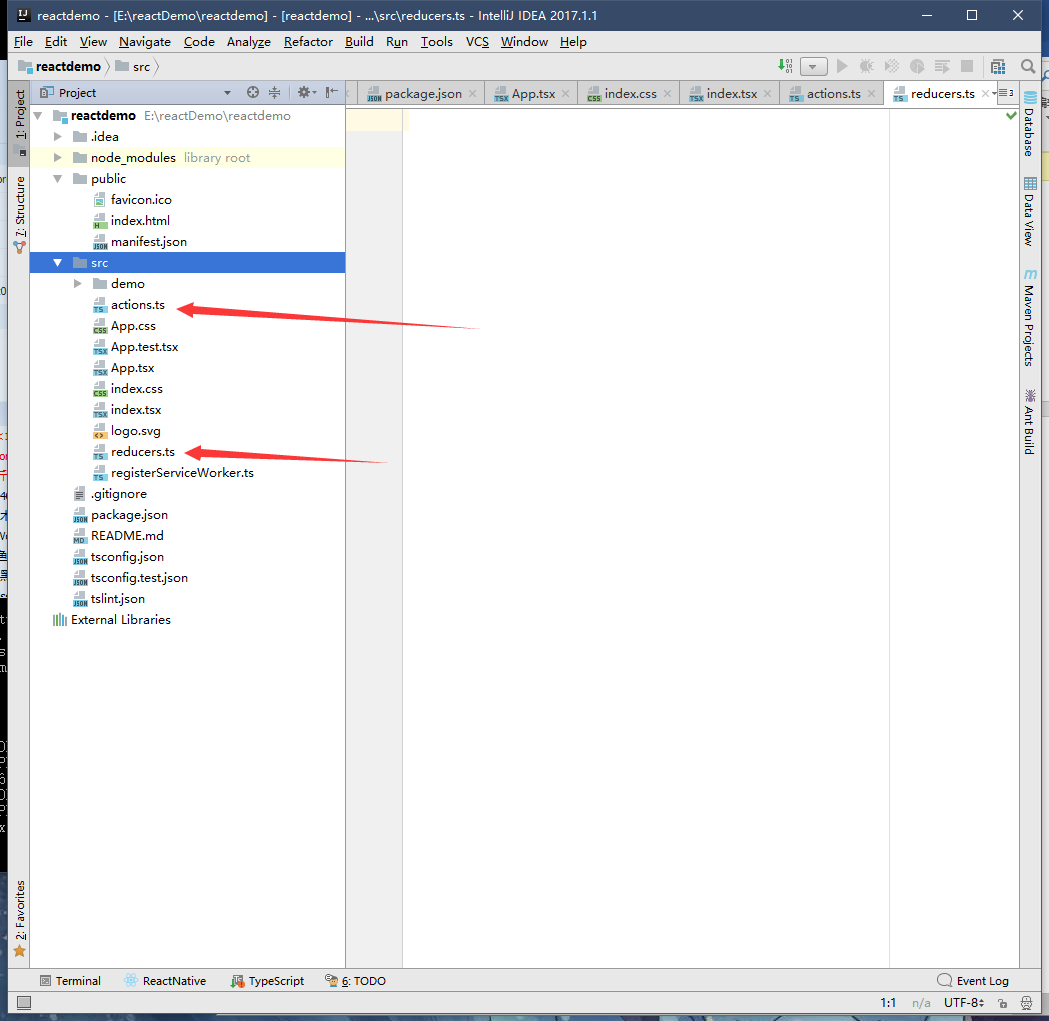
npm命令：npm install --save redux react-redux [react-router-redux@5.0.0-alpha.6](mailto:react-router-redux@5.0.0-alpha.6) history @types/redux @types/react-redux @types/react-router-redux

注意：react-router-redux需要安装5.0.0-alpha.6以上版本，并且需要安装history才可以正常运行

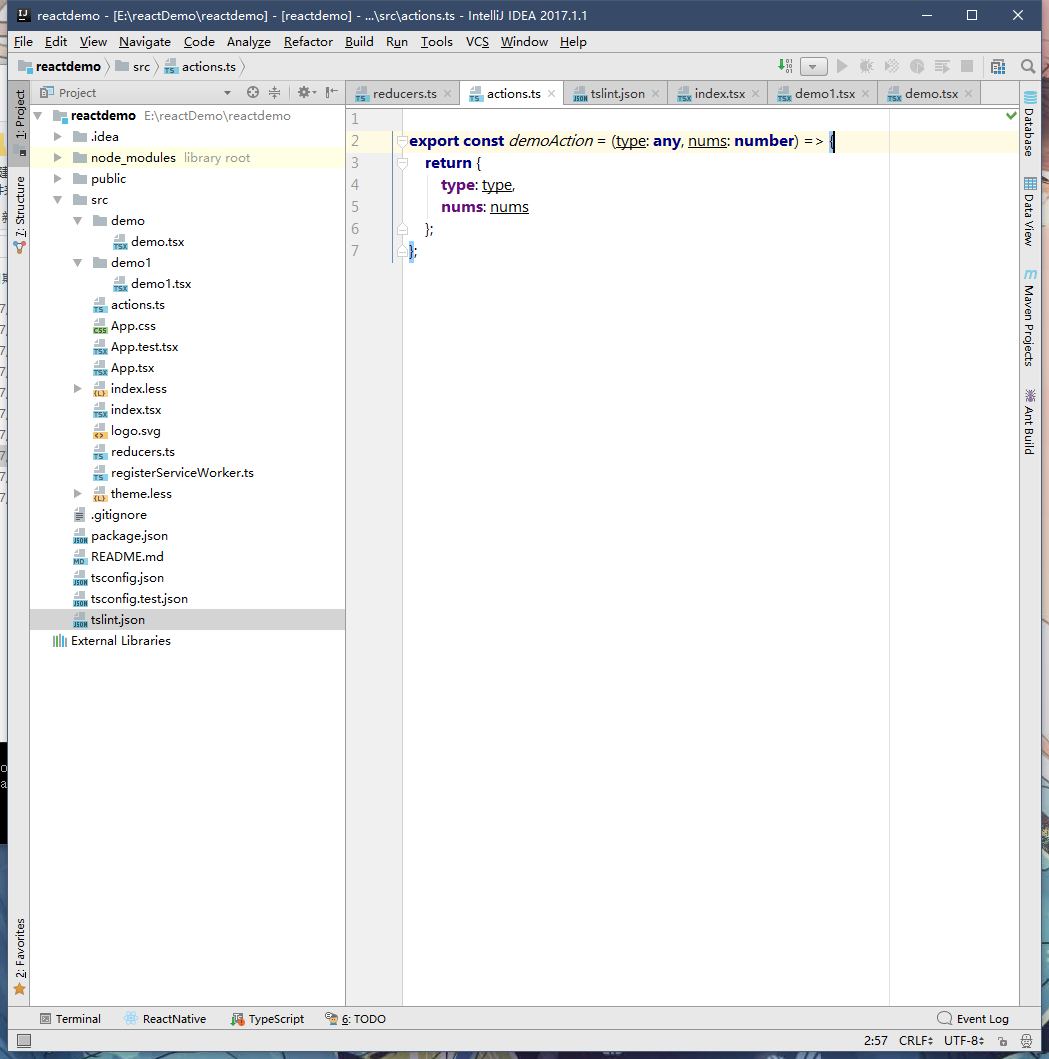


## 使用

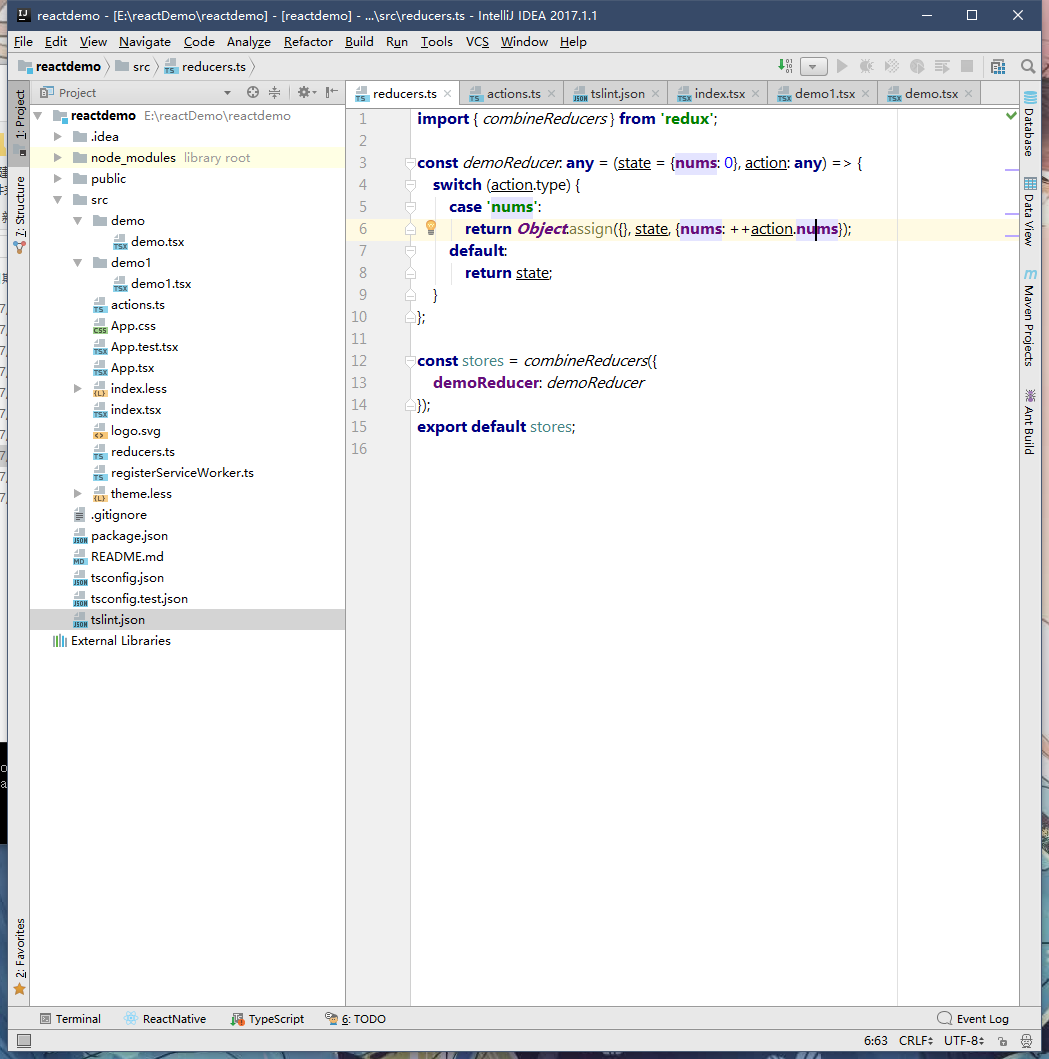
在src目录下新建actions.ts和reducers.ts：



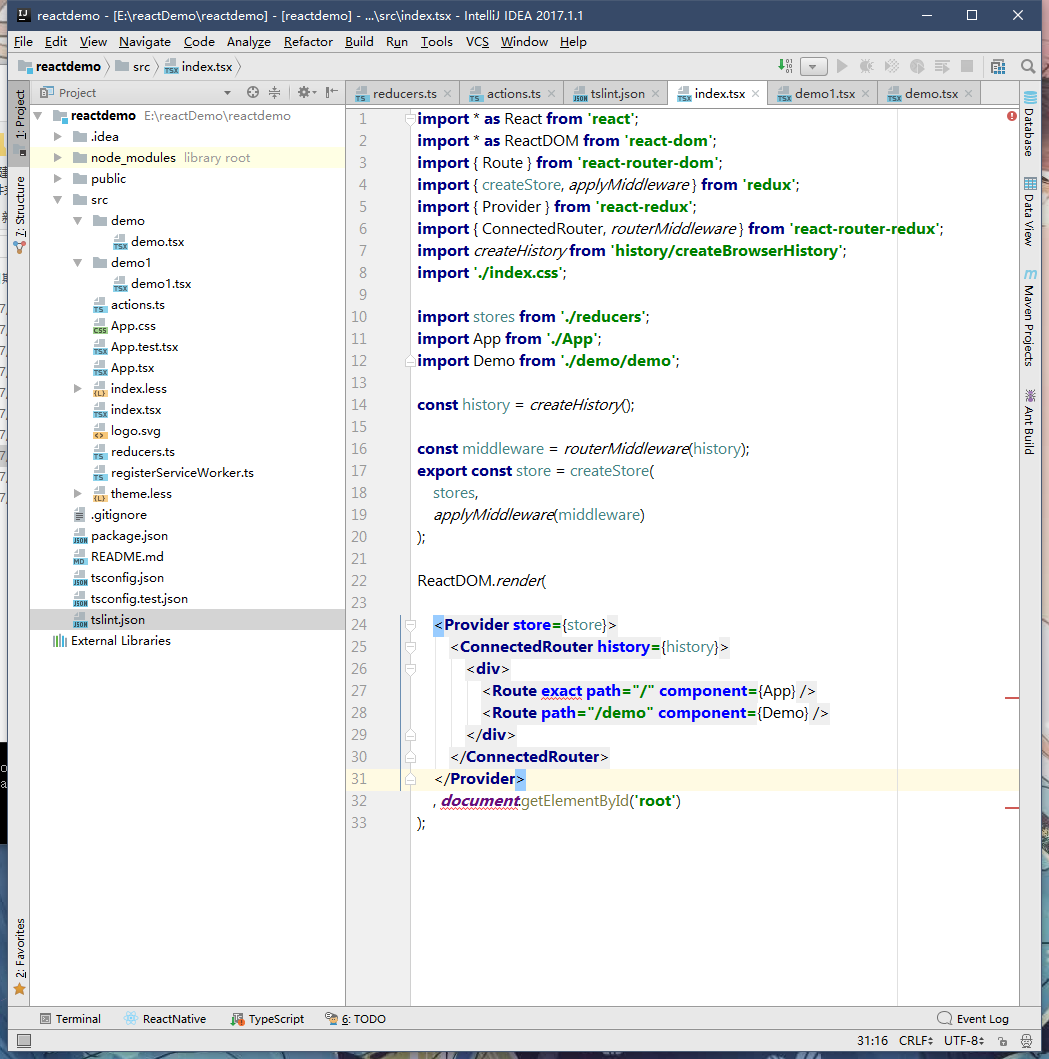
Actions.ts就是aciotn的合集，在action中我们写一个演示的action：



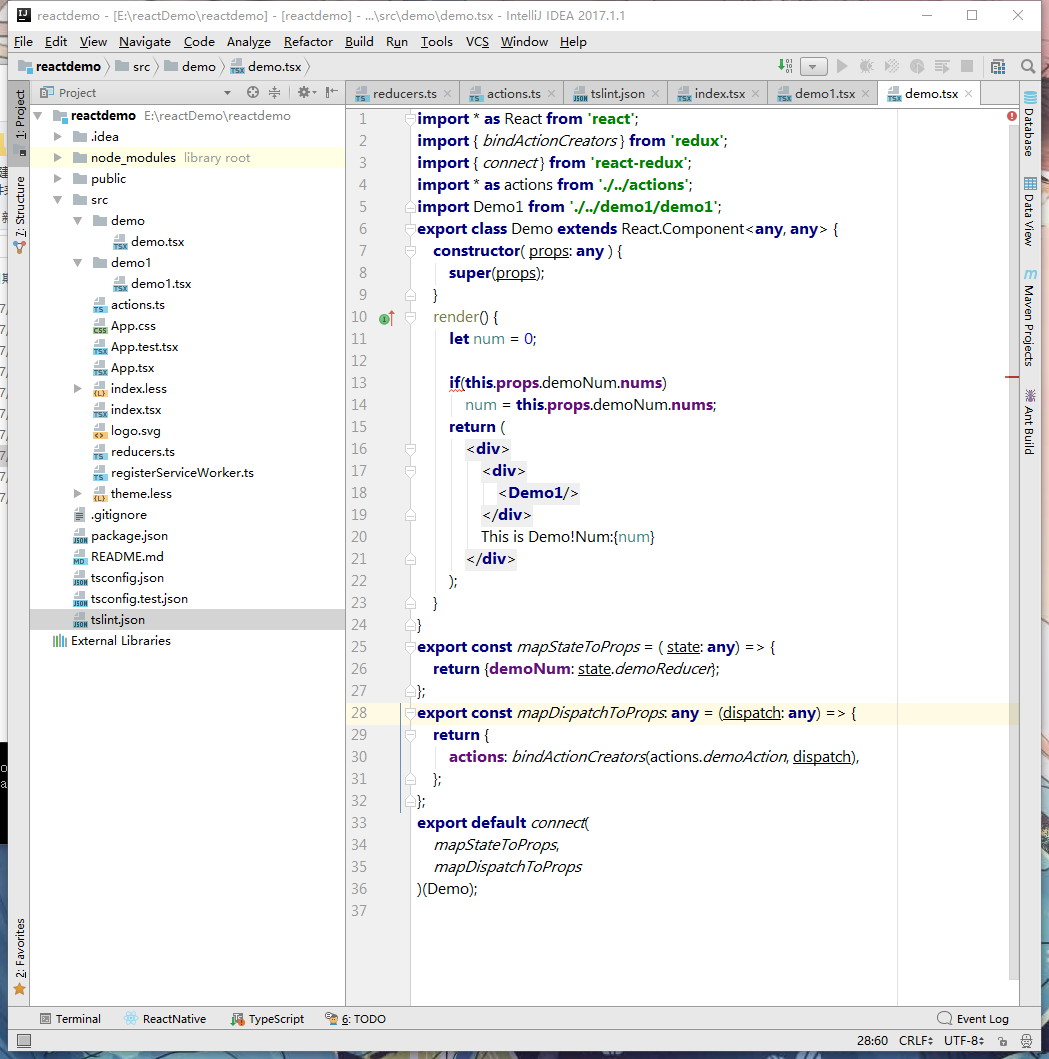
reducers.ts是reducer的合集，同样，我们写一个演示reducer：



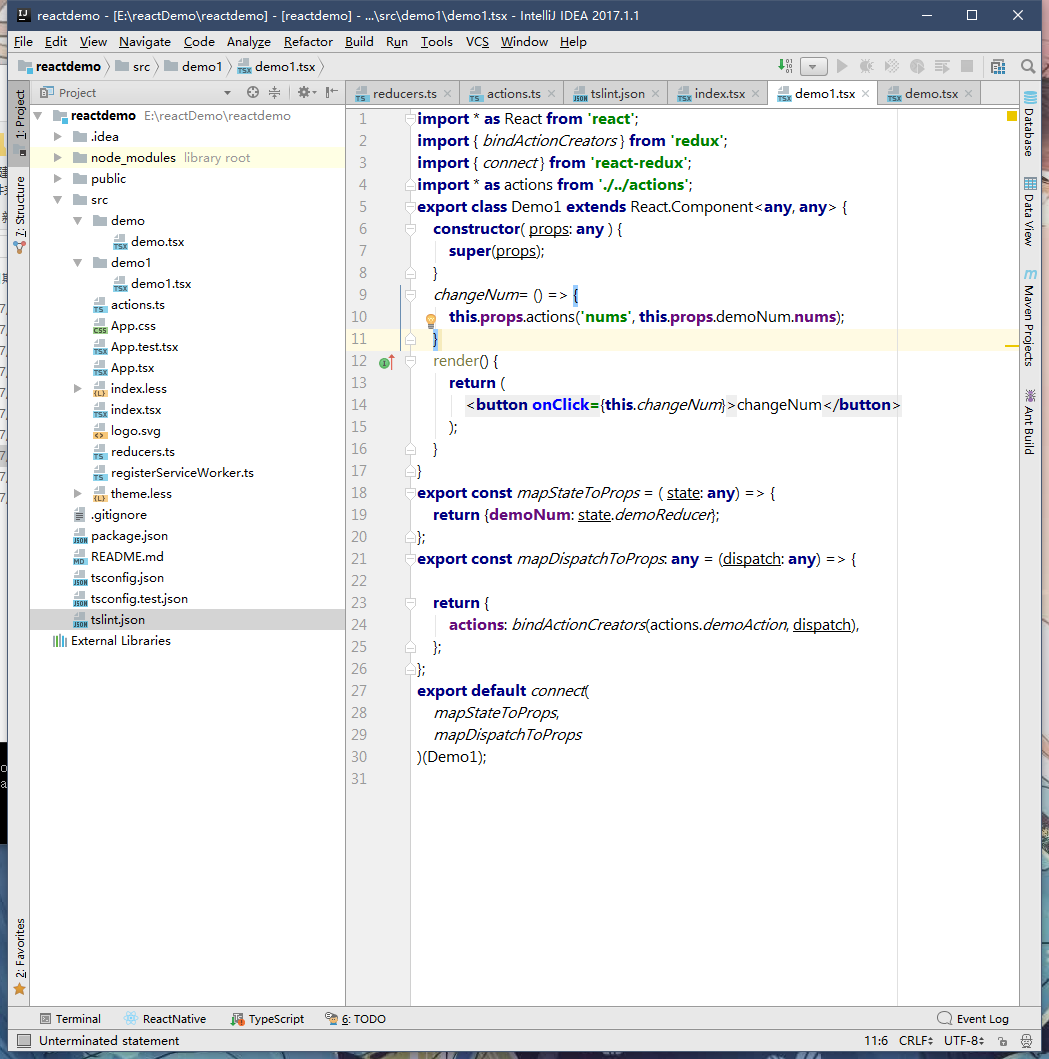
接下来，还需要修改我们的index.tsx



Demo.tsx:

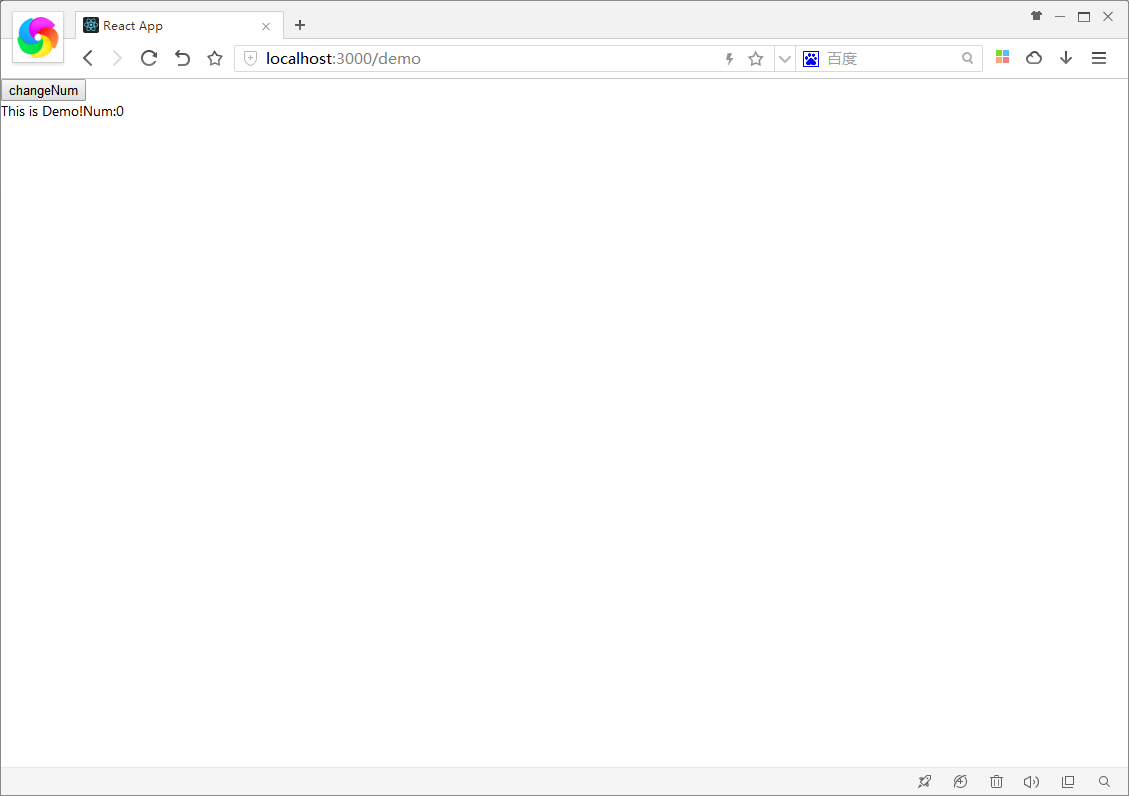


新增demo目录及demo1.tsx:

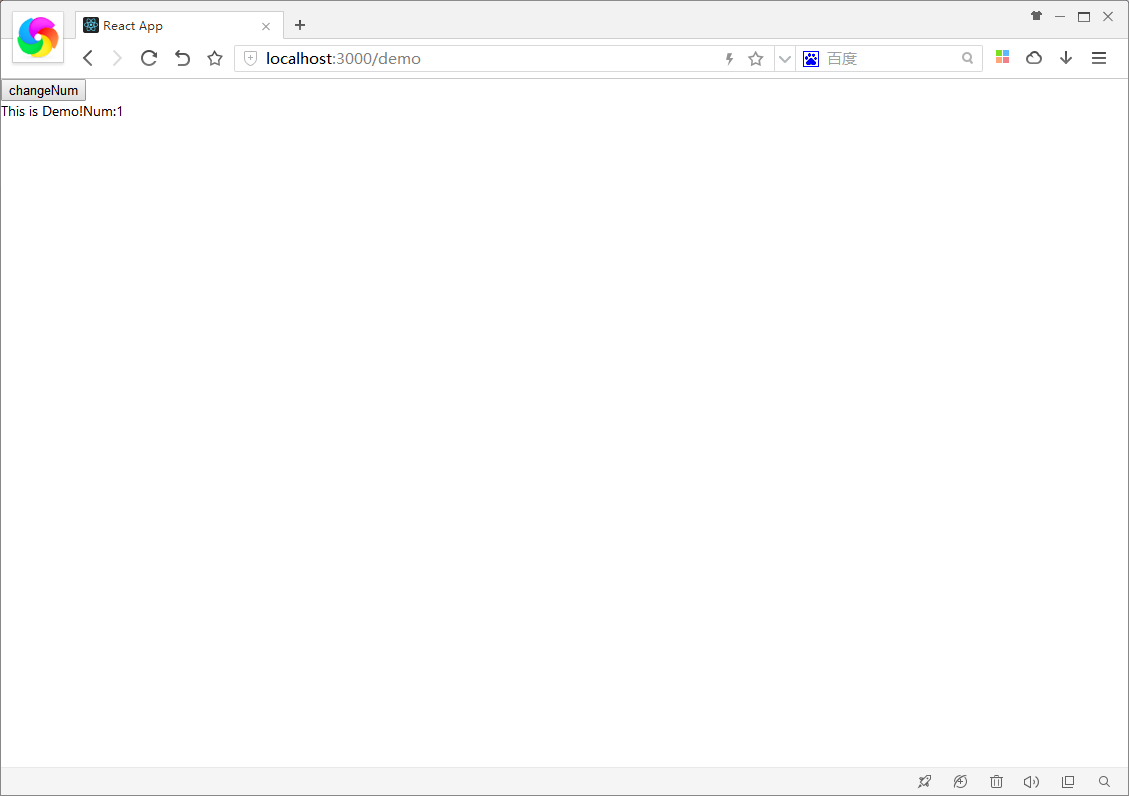


好了，这时，我们的两个组件（demo和demo1）已经可以通信啦

浏览器切换到我们的demo路径



点击按钮，可以看到demo组件的num已经变化：



组件通信已完成，细节先不补了

使用ant design做UI库：

# 小记

React,onclick使用：onclick={this.methodName.bind(event,param)}