

Angular4.x get post 以及 jsonp 请求数据

主讲教师：(大地)

合作网站: www.itying.com (IT 营)
合作网站: www.ionic.wang (Ionic 中文网)

目录

| | |
|--|---|
| 一、app.module.ts 注册 HTTP JSONP 服务 | 1 |
| 二、通过 Http、Jsonp 请求数据、不用 RxJs..... | 2 |
| 三、通过 Http、Jsonp 请求数据、使用 RxJs。 | 4 |

一、app.module.ts 注册 HTTP JSONP 服务

1.引入 HttpClientModule 、JsonpModule

普通的 HTTP 调用并不需要用到 JsonpModule，不过稍后我们会演示对 JSONP 的支持，所以现在加载它，免得再回来浪费时间。

```
import { HttpClientModule, JsonpModule } from '@angular/http';
```

2.HttpModule 、JsonpModule 依赖注入

```
@NgModule({
  declarations: [
    AppComponent,
    HomeComponent,
    NewsComponent,
    NewscontentComponent
  ],
  imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule,
    JsonpModule,
    AppRoutingModule
  ],
  providers: [StorageService, NewsService],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

二、通过 Http、Jsonp 请求数据、不用 RxJs

RxJS 是一种针对异步数据流编程工具，或者叫响应式扩展编程；不管如何解释 RxJS 其目标就是异步编程，Angular 引入 RxJS 为了就是让异步可控、更简单。

app.module.ts 引入并且依赖注入完成以后，在需要用到的地方执行下面操作。

使用 Http、Jsonp:

1、在需要请求数据的地方引入 Http

```
import {Http,Jsonp} from "@angular/http";
```

2、构造函数内申明:

```
constructor(private http:Http,private jsonp:Jsonp) { }
```

3、对应的方法内使用 http 请求数据

```
this.http.get("http://www.phonegap100.com/appapi.php?a=getPortalList&catid=20&page=1")
    .subscribe(
        function(data){
            console.log(data);
        },function(err){
            console.log('失败');
        }
    );
```

```
this.jsonp.get("http://www.phonegap100.com/appapi.php?a=getPortalList&catid=20&page=1&callback=JSONP_CALLBACK")
    .subscribe(
        function(data){
            console.log(data);
        },function(err){
            console.log('失败');
        }
    );
```

使用 Post

1. 引入 Headers 、 Http 模块

```
import {Http,Jsonp,Headers} from "@angular/http";
```

2. 实例化 Headers

```
private headers = new Headers({'Content-Type': 'application/json'});
```

3.post 提交数据

```
this.http
  .post('http://localhost:8008/api/test',
    JSON.stringify({username: 'admin'}), {headers:this.headers})
    // .toPromise()
    .subscribe(function(res){
      console.log(res.json());
    });
```

三、通过 Http、Jsonp 请求数据、使用 RxJs。

需要用到请求数据的地方引入 Http 模块 Jsonp 模块，以及 rxjs

RxJS 是一种针对异步数据流编程工具，或者叫响应式扩展编程；可不管如何解释 RxJS 其目标就是异步编程，Angular 引入 RxJS 为了就是让异步可控、更简单。

大部分 RxJS 操作符都不包括在 Angular 的 **Observable** 基本实现中，基本实现只包括 Angular 本身所需的功能。

如果想要更多的 RxJS 功能，我们必须 **导入**其所定义的库来扩展 **Observable** 对象， 以下是 **这个**模块所需导入的所有 RxJS 操作符：

1、引入 Http、Jsonp、RxJs 模块

```
import {Http,Jsonp} from "@angular/http";  
import {Observable} from "rxjs";  
import "rxjs/Rx";
```

你可能并不熟悉这种 `import 'rxjs/Rx'` 语法，它缺少了花括号中的导入列表：`{...}`。

这是因为我们并不需要操作符本身，这种情况下，我们所做的其实是导入这个库，加载并运行其中的脚本，它会把操作符添加到 `Observable` 类中。

2、构造函数内申明：

```
constructor(private http:Http,private jsonp:Jsonp) { }
```

3、get 请求

```
this.http.get("http://www.phonegap100.com/appapi.php?a=getPortalList&catid=20&page=1")  
  .map(res => res.json()) .subscribe(  
    function(data){  
      console.log(data);  
    }  
  );
```

4、Jsonp 请求

```
this.jsonp.get("http://www.phonegap100.com/appapi.php?a=getPortalList&catid=20&page=1&callback=JSONP_CALLBACK")  
  .map(res => res.json()) .subscribe(  
    function(data){  
      console.log(data);  
    }  
  );
```

```
}  
);
```

`http.get` 方法中返回一个 `Observable` 对象，我们之后调用 RxJS 的 `map` 操作符对返回的数据做处理。