# Angular+SpringBoot的Http通信一

# 基础环境

* 1. SpringBoot 2.2.4
  2. Angular 7.+

# 实现方式思路

2.1 Angular 前端封装http请求的服务

（官方建议这样做，自己封装而不是直接使用）

2.2 SpringBoot需要创建跨域访问的配置类，这样才能正常的请求到数据

# 3.Angular前端发送http请求

## 3.1 引入HttpClientModule模块

|  |
| --- |
| **在app.module.ts文件中引入HttpClientModule模块** |
| import { HttpClientModule } from '@angular/common/http';  @NgModule({  imports: [  BrowserModule,  AppRoutingModule,  HttpClientModule  ],  }) |

## 3.2 自定义封装http服务

|  |
| --- |
| **自定义命令 ng g service common/ApiService** |
| import { Injectable } from '@angular/core';  import { HttpClient } from '@angular/common/http';  import { Observable } from 'rxjs';  @Injectable({  providedIn: 'root'  })  export class ApiServiceService {  //0. 这个apiUrl作为基础的默认请求前缀！很关键  apiUrl = 'http://127.0.0.1:8080/api/mysql/';  //1.将原来的httpClient注入进来，方便我们二次封装  constructor(private httpClient:HttpClient) { }  //1.自定义的第一个get方法  // 返回值类型是一个 可订阅流  // 里面的url是进行拼接的  get(url:string,option?:any):Observable<any>{  const url\_use = this.apiUrl+url;  return this.httpClient.get(url\_use,option);  }    } |

## 3.3 在组件中注入自定义的http服务

|  |
| --- |
| **在组件中的constructor方法中将自定义的服务注入进来即可** |
| constructor( private apiService:ApiServiceService ){} |

## 3.4 在组件中使用自定义的http服务方法

|  |
| --- |
| //4.使用自定义的get方法来进行请求的操作  const url = '/queryAllUsers\_m';  this.apiService.get(url).subscribe( // this.apiService就是注入进来的服务对象  (res)=>{ // 这个res就是获取到的返回值  console.log(res);  },  (error:HttpErrorResponse)=>{ // 这个error就是报错时的错误  console.log('An error has occurred！'+ error.message);  }  ); |

# 4.SpringBoot服务端接收请求并处理

## 4.1 创建跨区请求的配置类（关键）

|  |
| --- |
| @Configuration  public class CoresConfig implements WebMvcConfigurer {  @Override  public void addCorsMappings(CorsRegistry registry) {  //设置允许跨域的路径  registry.addMapping("/\*\*")  //设置允许跨域请求的域名  //也可以指定域名 .allowedOrigins("http://192.168.0.0:8080","http://192.168.0.1:8081")  .allowedOrigins("\*")  //是否允许证书 不再默认开启  .allowCredentials(true)  //设置允许的方法  .allowedMethods("\*")  //跨域允许时间  .maxAge(3600);  }  } |

## 4.2 开发controller层的请求处理操作（常规操作）

|  |
| --- |
| @RestController  @RequestMapping("/api/mysql")  public class UserController {  @Autowired  private UserService service;  @GetMapping("/queryAllUsers\_m")  public List<User> queryAllUsers\_m(){  return service.queryAllUsers\_m();  }  } |

## 4.3 正常开发service层，mapper等，不在赘述

# 5.关于Angular 中 Http服务的封装

本文中只是简单的介绍了一个get的最简单的例子，更多的更全面的封装，需要结合官网文档进行进一步的研究和操作