## 一．springcloud安装配置

https://blog.csdn.net/hu\_dongyang/article/details/80708888

## 二．angula服务

<Http://a.itying.com/api/productlist>

1. 安装

ng g service servicename

Or ng g service /目录/servicename

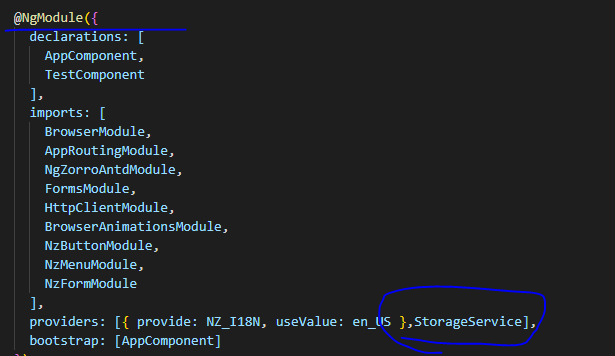
1. imort 依赖

在中引入

//引入 配置服务

import  {StorageService} from './services/storage.service';

1. 在providers中声明

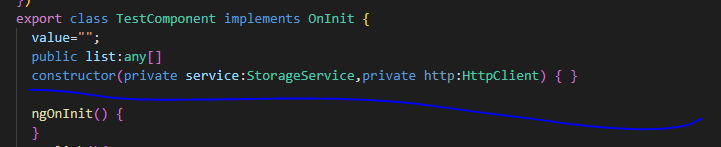


1. 在组件中使用

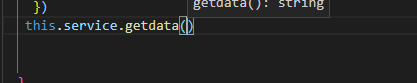
引入配置



依赖注入



调用服务中方法



## 接口请求

跨域配置：

### HttpClient

//请求放于服务

#### Get

    let url="http://a.itying.com/api/productlist";

    this.service.getrequest(url).subscribe(data=>{

       console.log(data)

       this.list=data['result']

    })

服务

  getrequest(url){

    // let url="http://a.itying.com/api/productlist";

    //

      return this.http.get(url)

  }

#### Post

 var url = "/api/angulartest";

    this.service.postrequest(url).subscribe(response => {

      console.log(response);

    });

服务

  postrequest(url){

    const httpOptions = {

      headers: new HttpHeaders({

         'Content-Type': 'application/json'

        })

    };

#### JsonP

### 第三方模块axios

Axios 使用文档

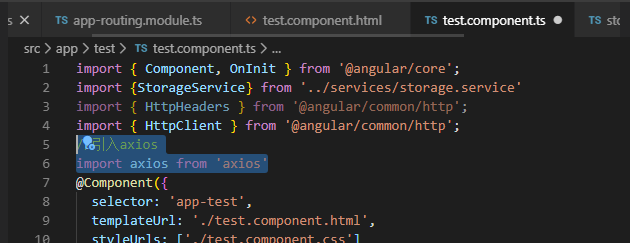
<https://www.kancloud.cn/yunye/axios/234845>

A：安装 npm install axios --save

B: 引入

//引入axios

import axios from 'axios'



C：使用

    axios.get('/user?ID=12345')

  .then(function (response) {

    console.log(response);

  })

  .catch(function (error) {

    console.log(error);

  });

法2

    axios({

      method: ’get’,

      url: url,

    }).then(res=>{

      console.log(res)

    }).catch(err=>{

      console.log(err)

    });

    }

Post 封装

 axiosPost(api){

      return new Promise((resolve,reject)=>{

              // 发送 POST 请求

              axios({

                method: 'post',

                url: '/user/12345',

                data: {

                  firstName: 'Fred',

                  lastName: 'Flintstone'

                }

              }).then(res=>{

                resolve(res)

              }).catch(err=>{

                reject(err)

              });

      })

    }

 axiosgetClick(){

    let url="http://a.itying.com/api/productlist";

    this.service.axiosGet(url).then(res=>console.log(res)).catch(err=>console.log(err))

  }

bug记录 laravel 安装没有开扩展报错

### angular跨域配置

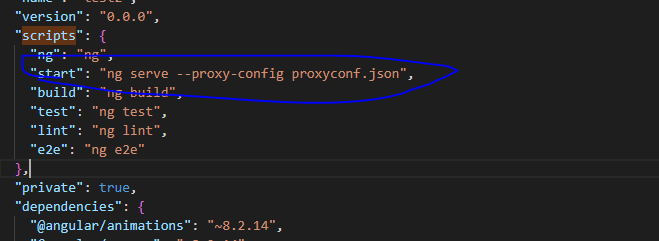
[https://angular.cn/guide/build#proxying-to-a-backend-server](https://angular.cn/guide/build" \l "proxying-to-a-backend-server)





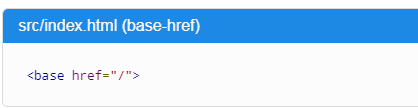


Angular 运行时读取你的代理

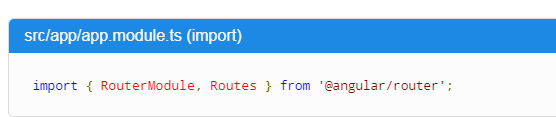


## 路由

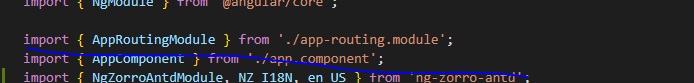
### 设置路由导航根路径

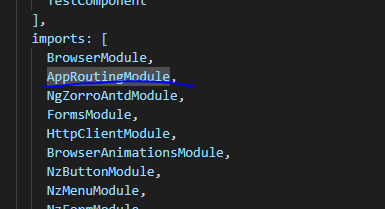


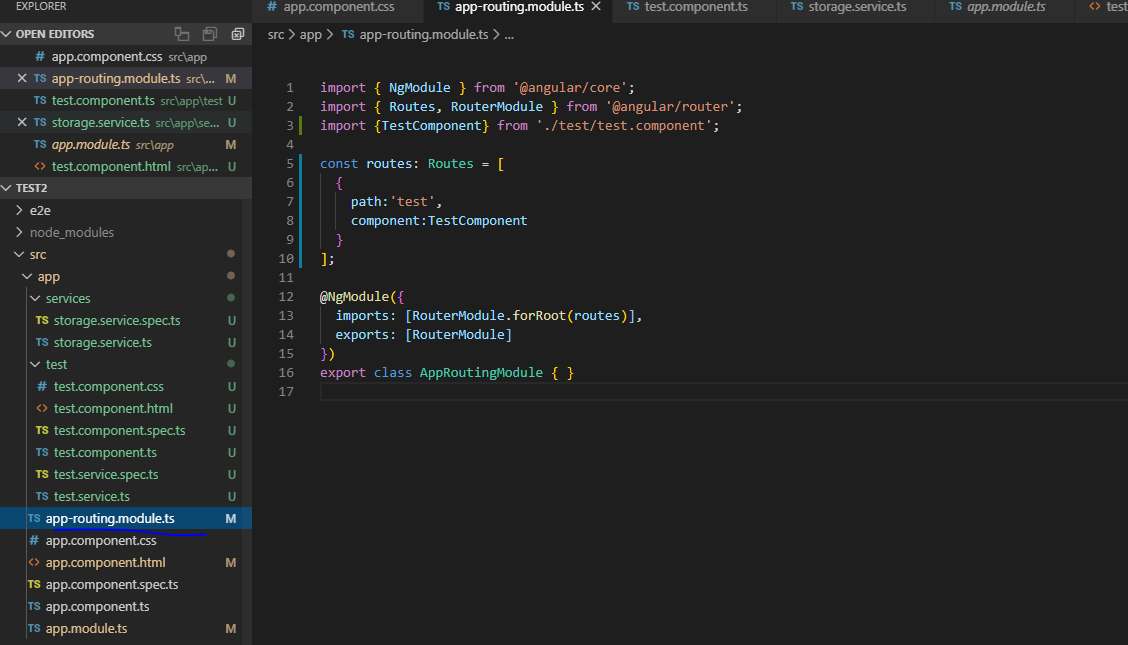
### 导入路由



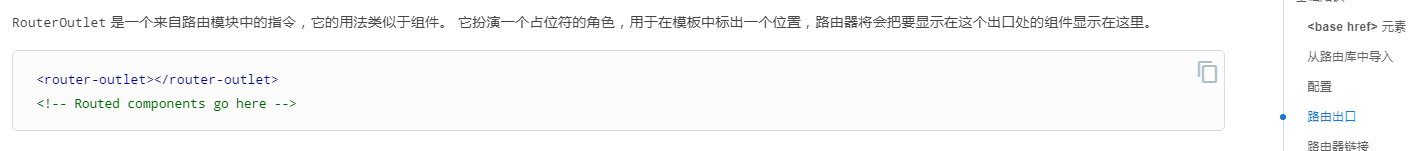
### 配置路由







### 路由出口



<[router-outlet](https://angular.cn/api/router/RouterOutlet)></[router-outlet](https://angular.cn/api/router/RouterOutlet)>

在模板中占位，将要显示的组件显示在这

### 路由连接

<[a](https://angular.cn/api/router/RouterLinkWithHref) [routerLink](https://angular.cn/api/router/RouterLink)="/crisis-center" [routerLinkActive](https://angular.cn/api/router/RouterLinkActive)="active">Crisis Center</[a](https://angular.cn/api/router/RouterLinkWithHref)> <[a](https://angular.cn/api/router/RouterLinkWithHref) [routerLink](https://angular.cn/api/router/RouterLink)="/heroes" [routerLinkActive](https://angular.cn/api/router/RouterLinkActive)="active">Heroes</[a](https://angular.cn/api/router/RouterLinkWithHref)>

### 定义路由通配符过滤非法路由

{ path: **'\*\*'**, component: 要跳转的组件的组件名}

当用户访问未定义的路由时自动跳转到预设的组件

### 路由参数

#### 7.1带参数的路由定义

{ path: 'hero/:id', component: HeroDetailComponent }

访问：

localhost:4200/hero/15

#### 从组件访问含参路由

<a [routerLink]="['/hero', hero.id]">

### 子路由

