# angular整合Java

由于node FFI模块安装失败未能解决，使用下面的方法代替。

## angular使用jsonp调用Java后台

### 1.HttpClientModule版本

1.1.对HttpClientJsonp模块的引入

在app.module.ts中引入HttpClientModule和HttpClientJsonpModule

import {HttpClientModule, HttpClientJsonpModule} from '@angular/common/http';

@NgModule({

declarations: [

AppComponent

],

imports: [

HttpClientModule,

HttpClientJsonpModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

1.2.在angular服务中使用

import { Injectable } from '@angular/core';

import {HttpClient} from '@angular/common/http';

import {catchError, timeout} from 'rxjs/operators';

import {throwError} from 'rxjs';

@Injectable({

providedIn: 'root'

})

export class HttpClientService {

urlRoot = 'http://localhost:8892/';

constructor(private http: HttpClient) { }

queryJsonpList() {

const url = this.urlRoot + 'testuser/jsonp/user/1';

return this.http.jsonp<any>(url, 'callback').pipe(

timeout(15000),

catchError((error) => this.handleSomeError(error))

);

}

private handleSomeError(error: any) {

if (error.error instanceof ErrorEvent) {

console.log(' error occurred:', error.error.message);

} else {

console.log(`${error.status}-${error.statusText}`);

}

return throwError(new Error('Server Error!'));

}

}

说明：实际的请求路径为：http://localhost:8892/testuser/jsonp/user/1?callback=ng\_jsonp\_callback\_0，其中callback作为参数的key，可以在后台的请求参数中获取。

1.3.调用service方法

clickHttpClientQueryJsonp() {

this.httpClient.queryJsonpList().subscribe(data => {

console.log(data.name);

});

}

### 2.HttpModule版本

2.1.引入Jsonp模块

在app.module.ts中引入HttpModule和JsonpModule

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

@NgModule({

declarations: [

AppComponent

],

imports: [

JsonpModule,

HttpModule

],

providers: [],

bootstrap: [AppComponent]

})

export class AppModule { }

2.2.在angular服务中使用

import { Injectable } from '@angular/core';

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

import {map, timeout} from 'rxjs/operators';

import {Observable} from 'rxjs';

@Injectable({

providedIn: 'root'

})

export class AppService {

times = 0;

urlRoot = 'http://localhost:8892/';

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

httpQueryJsonp() {

const url = this.urlRoot + 'testuser/jsonp/user/1' + '?callback=\_\_ng\_jsonp\_\_.\_\_req' + this.times + '.finished';

this.times++;

return this.jsonp.get(url).pipe(

map(this.handleData),

timeout(15000),

catchError((error) => this.handleSomeError(error))

);

}

private handleSomeError(error: any) {

if (error.error instanceof ErrorEvent) {

console.log(' error occurred:', error.error.message);

} else {

console.log(`${error.status}-${error.statusText}`);

}

return Observable.throw('Error!');

}

}

说明：url中callback作为参数的key，可以在后台的请求参数中获取。

2.3.调用service方法

clickHttpQueryJsonp() {

this.service.httpQueryJsonp().subscribe(data => {

console.log(data.name);

});

}

### 3.java后台接收和返回

@RequestMapping("/jsonp/user/{id}")

public void test(HttpServletResponse response, @RequestParam(name = "callback") String callbackName, @PathVariable(name = "id") Integer id) {

response.setContentType("text/javascript");

TestFracong test = testUserService.selectFracong(id);

Writer writer = null;

try {

writer = response.getWriter();

writer.write(callbackName + "(");

writer.write(JSON.toJSONString(test));

writer.write(");");

} catch (IOException e) {

logger.error("jsonp Error:" + JSON.toJSONString(test), e);

} finally {

if (writer != null) {

try {

writer.close();

} catch (IOException e) {

logger.error("writer close error！", e);

}

writer = null;

}

}

}

Java调用so已经实现(demo链接<https://github.com/Angeliau/sqlite-jna-demo.git>)，后续参考上例，使用angular调用Java后台，完善GUI显示。