14-4 axios非同步處理更新與刪除作業

axios ajax Framework

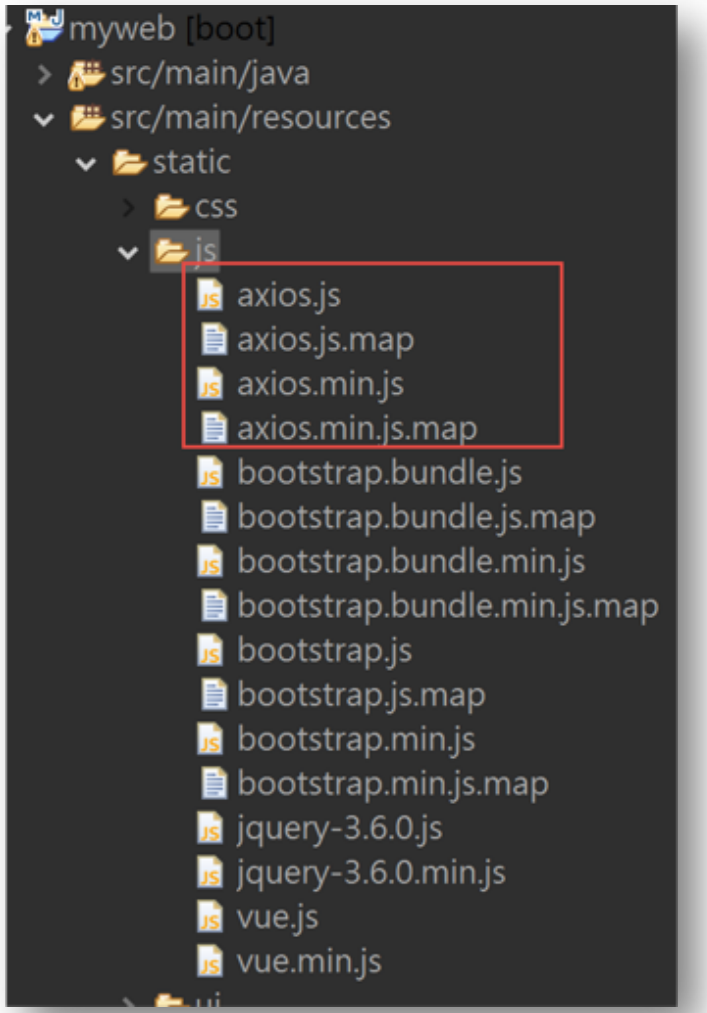
何謂axios

- axios 是一個Base on在 promise 的 HTTP Library。
- 可以採預GET/POST/PUT/DELETE進行非同步呼喚
- 輕量級ajax library

下載axios framework

配置axios

https://github.com/axios/axios



RESTful後端服務設計-更新客戶資料服務

採用RESTful架構設計服務

使用PutMapping設定consumes options

- 要求前端採用JSON內容傳遞
- 反序列化封裝資訊至JavaBean

使用JdbcTemplate進行資料異動作業

採用produces設定回應前端訊息為JSON文件格式

```
//接受傳遞進來的客戶資料 進行資料更新
//要求前端採用Request method-PUT 同時要求Request Header Content-Type:application/
@PutMapping(path="/customers/update/rawdata",
    produces="application/json;charset=UTF-8",
    consumes="application/json")
public Message customersUpdate(@RequestBody Customer customer,
        HttpServletResponse response) {
    String update="Update Customers set CompanyName=?,Address=?,"
            + "Phone=?,Country=? Where CustomerID=?";
    Message msg=new Message();
    //借助注入進來的JdbcTemplate
    try {
    int affect=jdbcTemplate.update(update,
            customer.getCompanyName(),
            customer.getAddress(),
            customer.getPhone(),
            customer.getCountry(),
            customer.getCustomerId()
            );
```

```
    //客戶被刪除了 還是更新成功 零筆
    if(affect>0) {
        msg.setCode(200);
        msg.setMsg("客戶資料更新成功");
        //預設Http status code 200
    }else {
        //Http status code=400
        response.setStatus(400);
        msg.setCode(400);
        msg.setMsg("客戶資料更新不到 資料不存在");
    }
```

Server Resource配置-控制器註冊與注入服務位址

自訂服務位址的resource file

控制器註冊自訂Resource 與注入項目

採用thymeleaf 嵌入服務位址為前端JS Variable

- 轉換成前端Vue資料模組
- 提供前端axios採用非同步呼喚使用

```
services.properties
outside.ubike.service=https://tcgbusfs.blob.core.windows.net/dotapp/youbike/v2/yc
service.customers.update=http://localhost:8080/tibamecustomers/update/rawdata
service.customers.delete=http://localhost:8080/tibamecustomers/delete/key/%s/rawc
```

```
1 package com.tibame.controller;
2
3 import java.io.IOException;
31 //POJO(Plain Old Java Object)
32 //客戶CRUD控制器
33 //注入一個依賴關係(DI-Dependency Injection)的JavaBean 進行客戶資料新增用
34 //註冊特定的properties file
35 @Controller
36 @PropertySource(value= {"classpath:services.properties"})
37 @RequestMapping(path="/tibamecustomers")
38 public class TibameCustomersController {
```

```
//注入資源檔項目
@Value("${service.customers.update}") //使用Spring EL
private String customersUpdateService;
//刪除服務位址
@Value("${service.customers.delete}") //使用Spring EL
private String customersDeleteService;
```

```
<script th:inline="javascript">
    //global變數 注入thymeleaf 運算變數內容 變成JavaScript在網頁中的敘述
    var country=/*[[${country}]]*/ ''
    var message=/*[[${message}]]*/ ''
    //Json String raw data變成JavaScript array
    var data=/*[[${data}]]*/ []
    //更新客戶服務位址
    var updateService=/*[[${updateService}]]*/ ''
    //刪除客戶服務位址
    var deleteService=/*[[${deleteService}]]*/ ''
```

Client Side對話盒更新資料-axios ajax非同步呼喚後端服務

使用axios.put method進行服務呼喚

put(app.updateAPI,data=app.curCustomer,header={"Content-Type":"application/json"})

傳遞相對物件JS (JSON 物件)

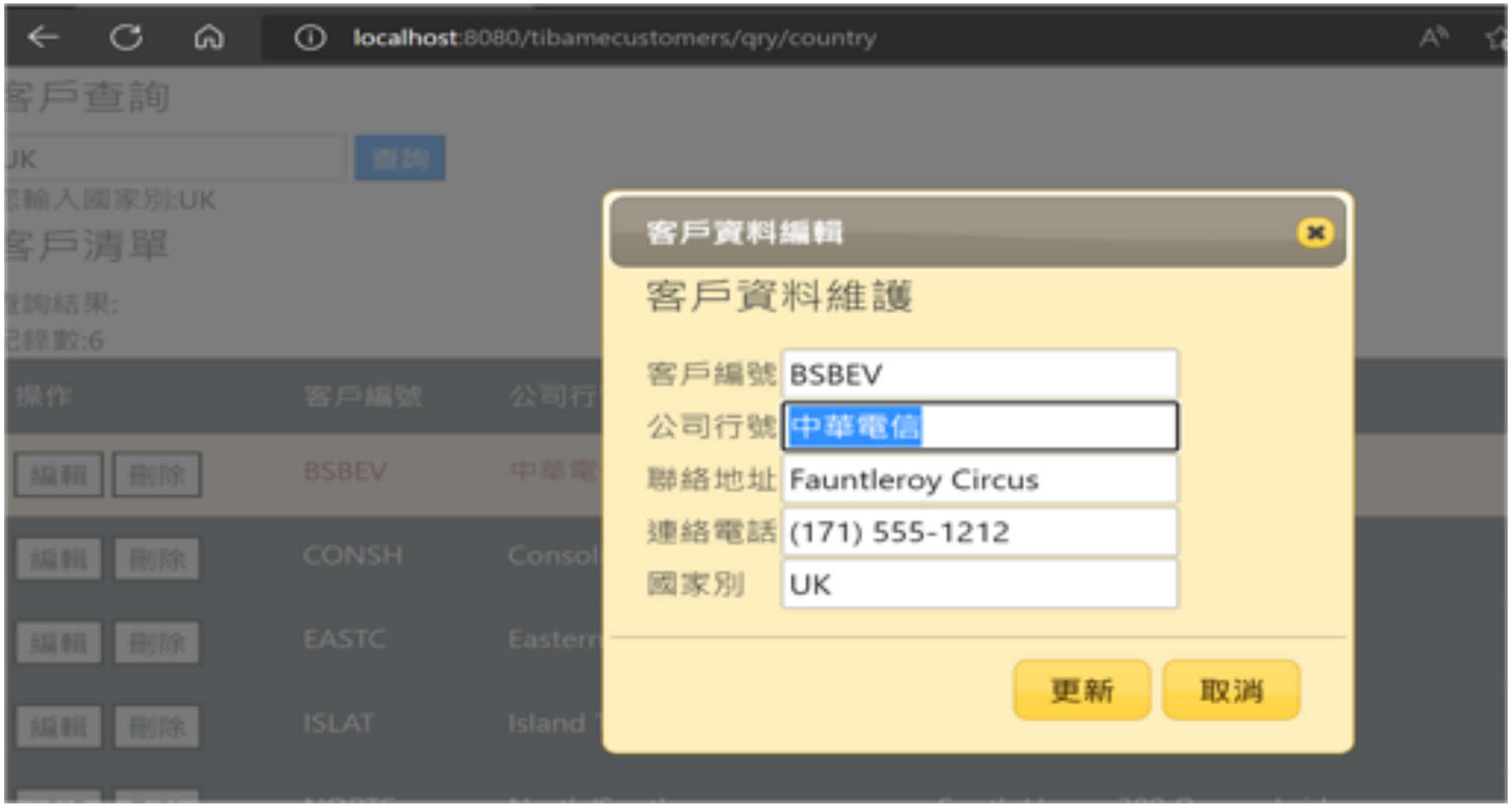
設定request Content-Type:application/json

使用then() 進行callback處理

```
//客製化按鈕(更精確)
buttons:[
    {
        text:'更新',
        click:function(){
            app.isAwait=true;
            //TODO 進行非同步處理 呼喚後端服務 進行資料更新
            console.log(app.updateAPI);
            //進行axios非同步處理
            //設定equest header-Content-Type:application/json
            //帶資料
            axios.put(app.updateAPI,data=app.curCustomer,
            header={"Content-Type":"application/json"})
            //success callback Http Status 2xx
            .then(res => {
                console.log(res)
                //取出資料
                app.alertMessage.text=res.data.msg;
                //圖
                app.alertMessage.image='../../images/success.png';
                app.isMessage=true;
                app.isWait=false;

            })
```

Client Side對話盒更新資料-axios ajax Demo



總結：14-4 axios非同步處理更新與刪除作業

學習到axios進行ajax非同步處理，呼喚後端服務，進行資料更新，下一個章節我們來探討Spring Security架構。