

13-1

使用JPA進行資料
維護應用-客戶新
增作業





第十三單元大綱

- 13-1 前端查詢畫面刻版設計
- 13-2 JavaScript與DOM基礎應用
- 13-3 JavaScript資料模組設計
- 13-4 單元測驗



採用同一個Action端點，進行第一次請求調用查詢表單頁面或者Postback進行查詢控制流程

使用@RequestMapping設定兩種一上的Http Request Method

注入HttpServletRequest物件進行傳送方法判斷

注入JdbcTemplate DAO物件進行後端資料庫查詢作業

```
TibameCustomersController.java
114     }
115
116
117     //依照國家別進行相關客戶查詢
118     //注入底層封裝前端所有資訊的HttpServletRequest(DI-Dependency Injection-依賴注入)
119     @RequestMapping(path= {"/qry/country"},
120                     method= {RequestMethod.GET, RequestMethod.POST})
121     public String customersQry(String country,
122                               HttpServletRequest request, Model model) {
```

```
public class TibameCustomersController {
    //注入客戶物件Customer Attribute注入
    //Autowired
    //private Customer customer;
    //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;

    //自動縫合 注入物件(使用型別認定-Spring Container)
    @Autowired
    private SQLServerDataSource mySqlDataSource;
    //Attribute注入JdbcTemplate
    @Autowired
    private JdbcTemplate jdbcTemplate; //實現DAO(Data Access Object)
```

Servlet api底層物件注入

- HttpServletRequest

使用getMethod()判斷請求方法

作為第一次請求或者postback進行查詢依據

```
//依照國家別進行相關客戶查詢
//注入底層封裝前端所有資訊的HttpServletRequest (DI-Dependency Injection-依賴注入)
@RequestMapping(path= {"/qry/country"},
    method= {RequestMethod.GET, RequestMethod.POST})
public String customersQry(String country,
    HttpServletRequest request, Model model) {
    //如何認定是第一次來 進行畫面調用 還是postback進行客戶查詢作業
    if(request.getMethod().equals("GET")) {
        //第一次來 調用畫面
        System.out.println("第一次來");
    } else if(request.getMethod().equals("POST")) {
        //進行查詢 使用JdbcTemplate (配合DataSource物件) 採用IoC注入控制反轉 直接反轉注入D
        try {
            System.out.println(jdbcTemplate.getDataSource().getConnection())
        } catch (SQLException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

對應Action回應的 旗標名稱在Thymeleaf資料夾中新增頁面

使用Visual Studio code進行編輯

使用<form> </form> 設計表單頁面

使用<input/> 設計文字輸入方塊(輸入國家別)

<input type= "submit" /> 設定交付觸發按鈕

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>客戶資料查詢</title>
</head>
<body>
  <fieldset id="panel">
    <legend>客戶查詢</legend>
    <form method="post">
      <div>
        <input type="text" name="country"/>
        <input type="submit" value="查詢"/>
      </div>
    </form>
  </fieldset>
</body>
</html>
```

使用外層元素進行封裝目的

- 可以透過CSS定義樣式

可以作為Vue進行掛載依據

進行版面區段管理

```
<body>  
  <fieldset id="panel">  
    <legend>客戶查詢</legend>  
    <form method="post">  
      <div>  
        <input type="text" name="country"/>  
        <input type="submit" value="查詢"/>  
      </div>  
    </form>  
  </fieldset>  
</body>
```



總結：13-1 前端查詢畫面刻版設計

透過MVC架構規劃一個前端查詢資料表單頁面，接下來我們探討如何設計Client Side程式碼(JavaScript)。

