



巨匠線上真人

Java Web OCE JWCD元件系統 開發認證

www.pcschoolonline.com.tw



巨匠線上真人

Java Web OCE JWCD元件系統開發認證

第十堂

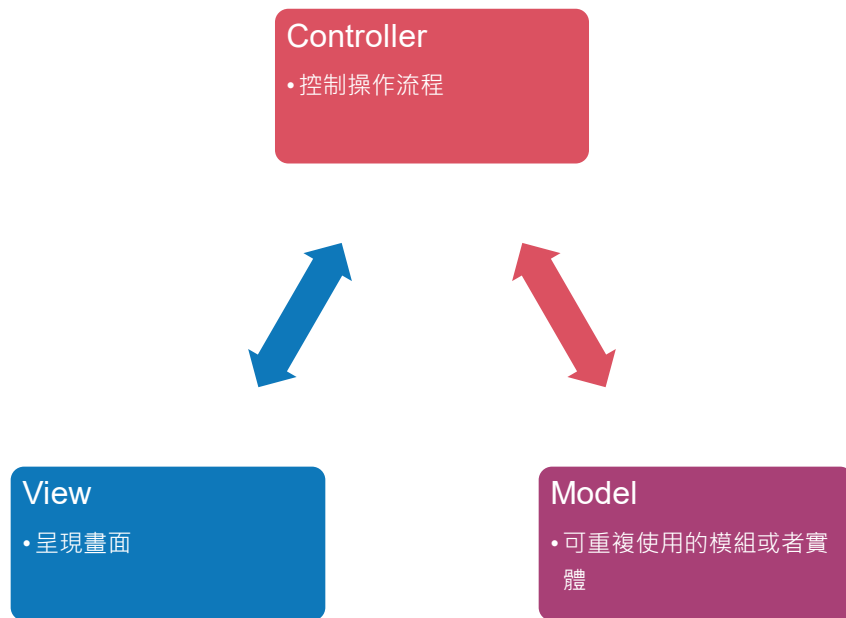
網站軟體工程-MVC應用

本堂教學重點

- ◆ 何謂MVC Pattern
- ◆ Servlet與JSP支援Java Web的定位與特性
- ◆ 使用Java Web Model2設計一個MVC操作
- ◆ Java Framework支援

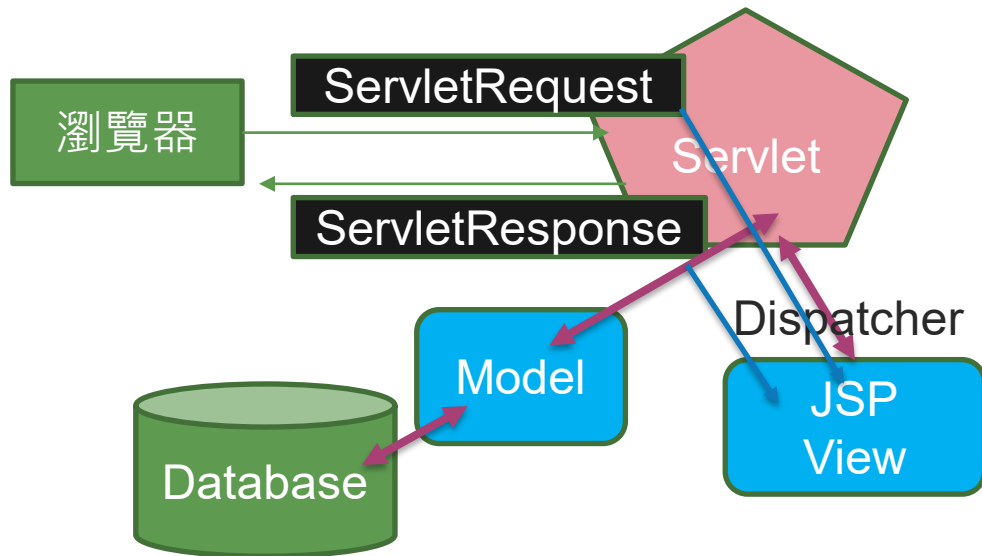
何謂MVC Pattern

- ◆ MVC模式 (Model-view-controller) 是軟體工程中的一種軟體架構模式 (Design Pattern) 。
- ◆ 把應用系統分為三個基本部分
 - ◇ 模型 (Model)
 - ◇ 視圖 (View)
 - ◇ 控制器 (Controller)
- ◆ 目的在於實現一種動態的程式設計，俾使後續對程式的修改和擴充簡化 (可擴充可維護性高)，並且使程式某一部分的重複利用。



Servlet與JSP支援Java Web的定位與特性

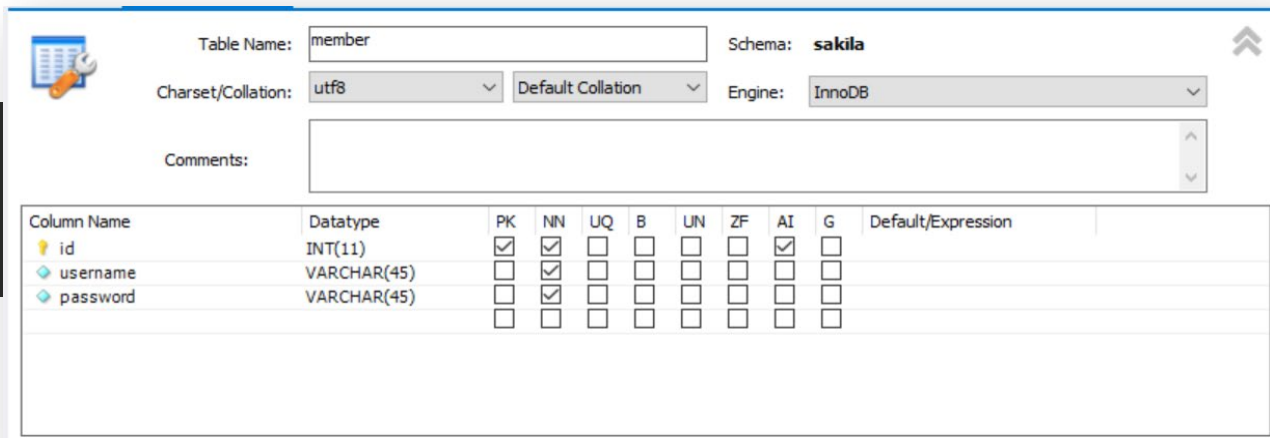
- ◆ Servlet API著重在Java程式碼轉寫，可以用來定義為與前端使用者界接的一個控制流程。並且透過Servlet呼叫可重複使用的Model進行邏輯與資料存取操作等。
- ◆ JSP強化在使用者介面設計，可以定義為View的實作部分。
- ◆ 我們不單獨使用一個JSP或者是一個Servlet直接完成一項操作，取其個別優勢，分工設計與整合，達到一個具有可維護性與彈性的系統設計架構。



使用Java Web Model 2設計一個MVC操作

- ◆ 設計一個會員註冊作業，需要的分工設計：
 - ◆ Static Page設計一個填寫註冊資訊的表單頁面。
 - ◆ Servlet設計接受表單頁面傳遞的註冊資訊，並且呼叫資料存取的DAO模組。完成註冊後進行分送到不同的View，其中包含有註冊不成功地分送View。
 - ◆ 設計JSP進行註冊結果的UI呈現。

準備了MySQL
範例資料庫sakila
規劃一個member Table



The screenshot shows the MySQL Workbench interface for creating or editing a table. The table name is 'member' in the 'sakila' schema. The charset is 'utf8' and the engine is 'InnoDB'. The table structure is defined as follows:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
username	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
password	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Member Table資料表

- ◆ 會員資料表設計在MySQL範例資料庫sakila中。
- ◆ SQL Script:

```
CREATE TABLE `member` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `username` varchar(45) CHARACTER SET latin1 NOT NULL,  
  `password` varchar(45) CHARACTER SET latin1 NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT  
CHARSET=utf8;
```

MySQL安裝與操作

◆ MySQL下載位址

◆ <https://dev.mysql.com/downloads/>

← → ↻ <https://dev.mysql.com/downloads/>

MySQL Community Downloads

- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository
- MySQL Community Server
- MySQL Cluster
- MySQL Router
- MySQL Shell
- MySQL Workbench
- C API (libmysqlclient)
- Connector/C++
- Connector/J
- Connector/NET
- Connector/Node.js
- Connector/ODBC
- Connector/Python
- MySQL Native Driver for PHP

MySQL Community Downloads

MySQL Community Server

General Availability (GA) Releases Archives

MySQL Community Server 8.0.19

Select Operating System:

Microsoft Windows

Looking for previous GA versions?

Recommended Download:

MySQL Installer for Windows

All MySQL Products. For All Windows Platforms. In One Package.

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

Windows (x86, 32 & 64-bit), MySQL Installer MSI

[Go to Download Page >](#)

Other Downloads:

General Availability (GA) Releases Archives

MySQL Installer 8.0.19

Select Operating System:

Microsoft Windows

Looking for previous GA versions?

Windows (x86, 32-bit), MSI Installer

8.0.19

18.6M

[Download](#)

(mysql-installer-web-community-8.0.19.0.msi)

MD5: 32043776cb2239db45fddaa86dc0ad61 | Signature

Windows (x86, 32-bit), MSI Installer

8.0.19

398.9M

[Download](#)

(mysql-installer-community-8.0.19.0.msi)

MD5: 1a882015da7fb93f20c4717e63b6817c | Signature



We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

規劃static Page填寫註冊資料

- ◆ 採用靜態頁面設計一個表單網頁。
- ◆ 用來填寫註冊資料。
- ◆ 並且規劃<form> action與method 屬性，用來傳送給指定的 Servlet 進行資料擷取與判斷等處理。

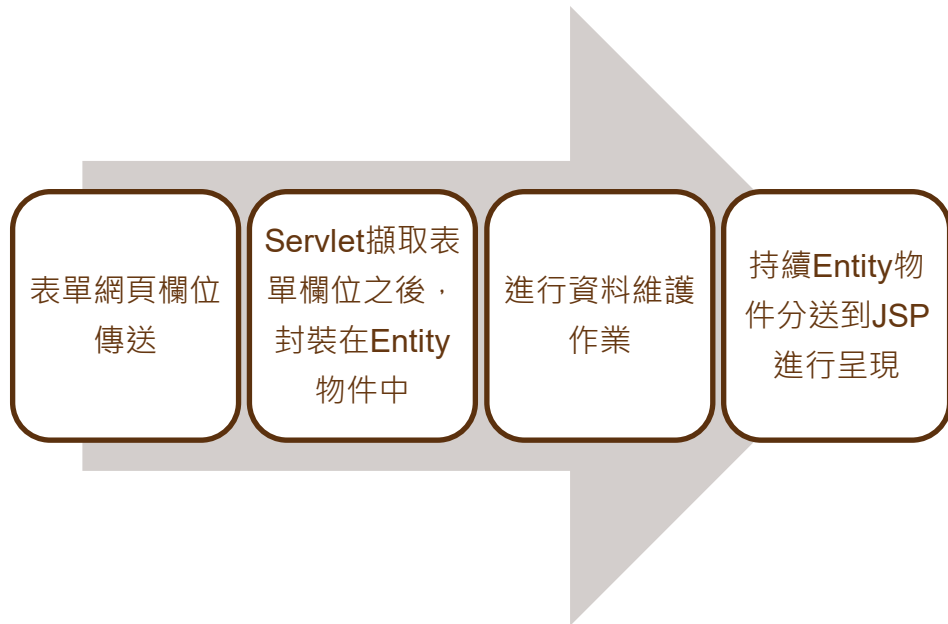
register.html

```
<body>
<fieldset>
  <legend>會員註冊</legend>
  <form method="post" action="">
    <div>使用者名稱</div>
    <input type="text" name="username" maxLength="20"/>
    <div>密碼</div>
    <input type="password" name="password" maxLength="20"/>
    <br/>
    <input type="submit" value="註冊"/>
  </form>
</fieldset>
</body>
```

Servlet URL

Entity Class規劃

- ◆ 表單欄位資訊，透過ServletRequest送到Servlet之後，將表單欄位封裝至一個Entity物件中，方便進行資料維護與狀態持續等作業。
- ◆ 規劃一個對應資料表欄位與對應表單網頁的表單欄位的Entity Class。
- ◆ Model實現的一部分。



Model-Entity Class規劃

Member.java

- ◆ 可以是JavaBean用在
<jsp:useBean>中，也可以程式化
用在Servlet中。
- ◆ 設定Class Property(setter/getter)
相對於Form Field name Attribute
設定。

```
package com.gjun.domain;  
//JavaBean -Entity Class設計  
public class Member implements java.io.Serializable{  
    private String username;  
    private String password;  
    public String getUsername() {  
        return username;  
    }  
    public void setUsername(String username) {  
        this.username = username;  
    }  
    public String getPassword() {  
        return password;  
    }  
    public void setPassword(String password) {  
        this.password = password;  
    }  
}
```



Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
username	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
password	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

```
<form method="post" action="">  
    <div>使用者名稱</div>  
    <input type="text" name="username" maxLength="20"/>  
    <div>密碼</div>  
    <input type="password" name="password" maxLength="20"/>  
    <br/>  
    <input type="submit" value="註冊"/>  
</form>
```

Controller-Servlet規劃-擷取表單欄位

- ◆ Servlet作為Controller角色定位在：
 - ◆ 與前端介面的資訊互通，如擷取註冊資料。
 - ◆ 進行後端資料驗證邏輯流程，可交由可重複使用的DAO模組進行。
 - ◆ 進行會員註冊資料的維護，可交由設計資料存取的DAO模組進行。
- ◆ 狀態(範圍可視度)管理與持續設定。
- ◆ 最後完成相對畫面的派送作業。

```
@WebServlet("/RegisterServlet")
public class RegisterServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        //擷取表單欄位
        String username=request.getParameter("username");
        String password=request.getParameter("password");
        //建構Entity物件用來封裝表單欄位內容
        Member member=new Member();
        member.setUsername(username);
        member.setPassword(password);
        //進行資料驗證 是否使用者名稱重複
        //如果重複分派到一個View去進行提醒
        //進行會員資料儲存
        //儲存成功 則分派一個View進行註冊資訊狀態呈現
    }
}
```

```
<legend>會員註冊</legend>
<form method="post" action="RegisterServlet">
    <div>使用者名稱</div>
    <input type="text" name="username" maxLength="20"/>
    <div>密碼</div>
    <input type="password" name="password" maxLength="20"/>
    <br/>
    <input type="submit" value="註冊"/>
</form>
```

Model設計-設計DAO模式

- ◆ DAO Pattern設計模式針對資料庫存取規範進行設計。
- ◆ 採用介面配合泛型，規劃對資料紀錄維護作業方法抽象描述。
- ◆ 透過針對特定資料表存取的類別規劃，實作DAO進行，可讓資料存取方法具有一定的規範，可以維持向前或者向後的版本相容性。

Idao.java

```
package com.gjun.domain;

import java.sql.SQLException;

import javax.sql.DataSource;
//使用Generic設計，可以配合類別實作時設定相對的Entity class
public interface IDao<T> {
    public void setDataSource(DataSource datasource);
    //查詢單筆方法
    public T selectForObject(String sql, Object...args) throws SQLException;
    //新增作業
    public boolean insert(T object) throws SQLException;
}
```

Model設計-Member DAO實作

- ◆ 規劃一個存取member資料表的DAO類別，實現DAO介面的方法，並且配合規劃好的Entity類別進行泛型(Generic)設定。
- ◆ 泛型設定會對應到相對的方法參數或者回應值。

MemberDao.java

```
public class MemberDao implements IDao<Member> {  
    //attribute  
    private DataSource datasource;  
  
    //強迫注入DataSource物件(連接工廠)  
    @Override  
    public void setDataSource(DataSource datasource) {  
        this.datasource=datasource;  
    }  
}
```

注入應用系統需要事前設定的DataSource物件
用來提供DAO模組產生連接物件的來源

Model設計-查詢單筆方法實作

MemberDao.java

- ◆ 用來進行會員資料查詢，依照使用者名稱進行單筆查詢。
- ◆ 亦可用來進行是否重複註冊作業驗證。

```
@Override
public Member selectForObject(String sql, Object... args) throws SQLException {
    Member member=null;
    //判斷DataSource物件是否注入
    if(this.datasource==null) {
        throw new SQLException("尚未注入DataSource物件");
    }
    //透過DataSource物件要一條連接物件
    Connection connection=datasource.getConnection();
    //產生一個帶參數查詢的PreparedStatement物件
    PreparedStatement st=connection.prepareStatement(sql);
    //走訪參數進行內容設定
    for(int pos=0;pos<args.length;pos++) {
        st.setObject(pos+1,args[pos]);
    }
    //執行查詢 產生結果
    ResultSet rs=st.executeQuery();
    //判斷是否有記錄
    if(rs.next())
    {
        //查有記錄
        member=new Member();
        member.setUsername(rs.getString("username"));
        member.setPassword(rs.getString("password"));
    }
    connection.close();
    return member;
}
```

Model設計-新增資料實作

MemberDao.java

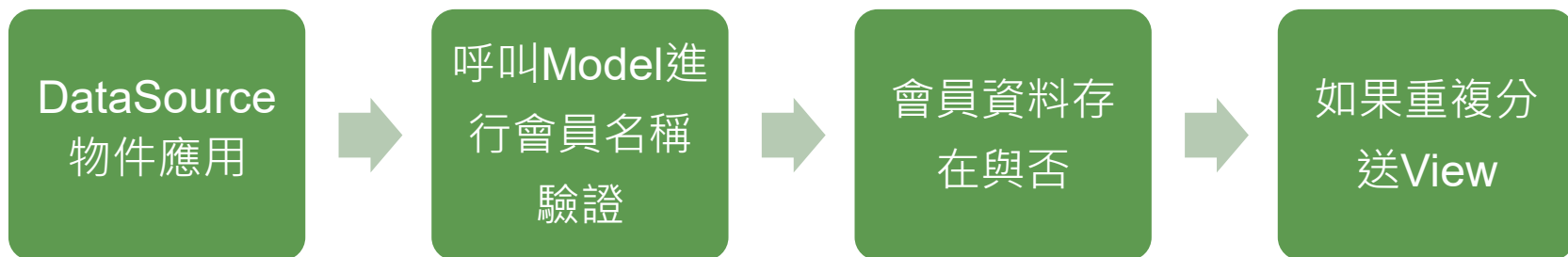
- ◆ 透過Geneic設定傳遞一個Member物件進行會員資料註冊新增作業。

```
@Override
public boolean insert(Member object) throws SQLException {
    //判斷DataSource物件是否注入
    if(this.datasource==null) {
        throw new SQLException("尚未注入DataSource物件");
    }
    //透過DataSource物件要一條連接物件
    Connection connection=datasource.getConnection();
    //產生一個帶參數查詢的PreparedStatement物件
    String sql="insert into member(username,password) values(?,?)";
    PreparedStatement st=connection.prepareStatement(sql);
    st.setString(1, object.getUsername());
    st.setString(2, object.getPassword());

    boolean result=false;
    //執行新增
    int r=st.executeUpdate();
    if(r>0) {
        result=true;
    }
    connection.close();
    return result;
}
```

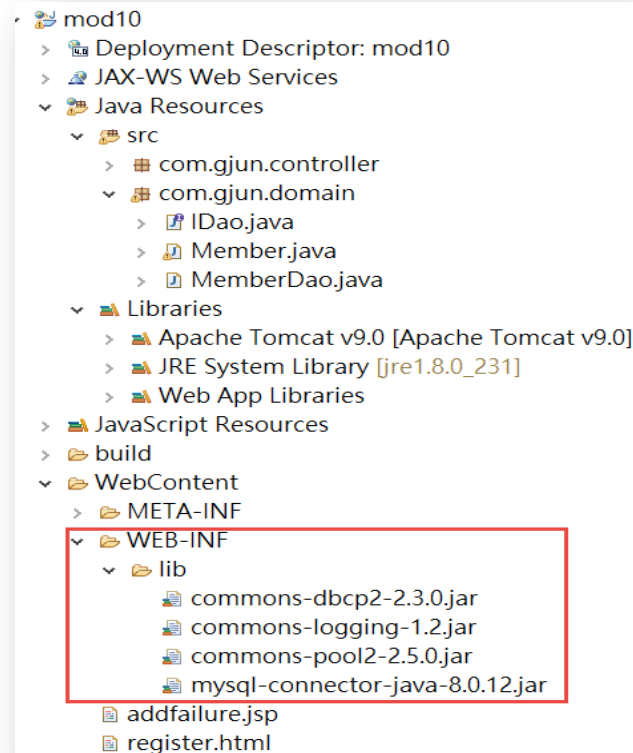

Controller-Servlet進行驗證作業

- ◆ Servlet處理會員註冊程序中，透過呼叫自訂Model進行會員註冊是否username重複。
- ◆ 如果重複則進行分送不同的View進行提醒。



資料庫存取API與DataSource API

- ◆ 存取MySQL Database API設定於網站系統中。
 - ◇ 在安裝好的MySQL資料夾C:\Program Files (x86)\MySQL\Connector J 8.0複製 Database API Jar File。
 - ◇ 放置網站WEB-INF資料夾中的lib資料夾中。
- ◆ 下載Apache DBCP Jar
- ◆ <https://jar-download.com/artifacts/org.apache.commons/commons-dbc2/2.3.0/source-code>



DataSource物件設計與自訂DAO整合

- ◆ DataSource為產生Connection(連接物件)工廠，一般可支援Connection Pooling進行效能調教。

DataSource 物件建立

- 配置URL
- 配置Driver Class
- 配置login帳密



自訂DAO物件建立

- 注入DataSource形成依存關係。

RegisterServlet.java

```
//建構DataSource物件
BasicDataSource dataSource=new BasicDataSource();
dataSource.setDriverClassName("com.mysql.cj.jdbc.Driver");
dataSource.setUrl("jdbc:mysql://localhost:3306/sakila?useSSL=false"
    + "&serverTimezone=UTC&characterEncoding=utf-8");
dataSource.setUsername("root");
dataSource.setPassword("1111");
//進行資料驗證 是否使用者名稱重複
IDao<Member> dao=new MemberDao();
//注入dataSource
dao.setDataSource(dataSource);
```

Controller-呼叫自訂模組驗證使用者名稱

- ◆ 驗證使用者名稱是否重複？
- ◆ 重複名稱則透過RequestDispatcher進行分送View。
- ◆ 透過ServletRequest進行狀態訊息參考，持續狀態到配分送的View中可以操作呈現。

自訂Model驗證會員名稱是否重複



使用
ServletRequest進行
狀態參考



透過
ServletRequest參
照出
RequestDispatcher
物件進行分送View
作業。

```
try {
    Member result=dao.selectForObject("select username,password from member where username=?",username);
    //判斷是否已經存在
    if(result!=null) {
        //名稱已經被註冊
        //透過ServletRequest參照出RequestDispatcher物件
        RequestDispatcher dispatcher=request.getRequestDispatcher("addfailure.jsp");
        //訊息狀態
        String msg=String.format("%s 使用者名稱已經被註冊!!", username);
        request.setAttribute("msg",msg);
        //分送
        dispatcher.forward(request, response);
    }else
    {
        //進行會員註冊作業(新增)
    }
} catch(SQLException ex) {
    RequestDispatcher dispatcher=request.getRequestDispatcher("addfailure.jsp");
    //訊息狀態
    String msg=ex.getMessage();
    request.setAttribute("msg",msg);
    //分送
    dispatcher.forward(request, response);
}
```

View設計-使用者名稱重複分送JSP設計

addfailure.jsp

- ◆ JSP View設計，用來呈現使用者名稱註冊重複，無法完成註冊作業。
- ◆ 經由RequestDispatcher.foward()分送進來。
- ◆ 使用EL取出原先的ServletRequest狀態參考資訊(字串內容)呈現。

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>註冊失敗</title>
</head>
<body>
<div style="font-size:30px;color:red">${msg}</div>
</body>
</html>
```

會員註冊使用者名稱重複Demo

← → ↻ 🏠 ⓘ localhost:8080/mod10/register.html

會員註冊

使用者名稱

eric

密碼

....

註冊

	id	username	password
▶	1	eric	1111
	2	linda	1111
	3	sam	1111
•	NULL	NULL	NULL

使用者名稱已經註冊



← → ↻ 🏠 ⓘ localhost:8080/mod10/RegisterServlet

eric 使用者名稱已經被註冊!!

Controller-完成新增會員

- ◆ Servlet控制程序中呼叫自訂DAO模組進行會員註冊作業。
- ◆ 註冊成功進行相對的View分送。



```
//判斷是否已經存在
if(result!=null) {
    //名稱已經被註冊
    //透過ServletRequest參照出RequestDispatcher物件
    RequestDispatcher dispatcher=request.getRequestDispatcher("addfailure.jsp");
    //訊息狀態
    String msg=String.format("%s 使用者名稱重複!!", username);
    request.setAttribute("msg",msg);
    //分送
    dispatcher.forward(request, response);
}else
{
    //進行會員註冊作業(新增)
    boolean rs=dao.insert(member);
    //透過ServletRequest參照出RequestDispatcher物件
    RequestDispatcher dispatcher=request.getRequestDispatcher("addok.jsp");
    //訊息狀態
    String msg=String.format("使用者 %s 完成註冊!!", username);
    request.setAttribute("msg",msg);
    //分送
    dispatcher.forward(request, response);
}
```

View設計-完成註冊JSP

- ◆ 使用EL取出requestScope範圍參考的attribute呈現

```
//進行會員註冊作業(新增)
boolean rs=dao.insert(member);
//透過ServletRequest參照出RequestDispatcher物件
RequestDispatcher dispatcher=request.getRequestDispatcher("addok.jsp");
//訊息狀態
String msg=String.format("使用者%s 完成註冊!!!", username);
request.setAttribute("msg",msg);
//分送
dispatcher.forward(request, response);
```

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>註冊成功</title>
8 </head>
9 <body>
10 <div style="color:blue;font-size:40px">${msg}</div>
11 </body>
12 </html>
```


會員註冊Demo

← → ↻ 🏠 ⓘ localhost:8080/mod10/register.html

會員註冊

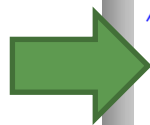
使用者名稱

allen

密碼

...

註冊



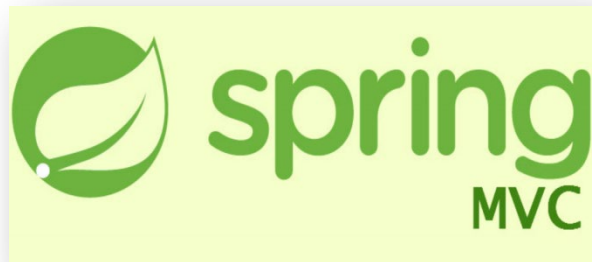
← → ↻ 🏠 ⓘ localhost:8080/mod10/RegisterServlet

使用者 allen 完成註冊!!

Server Side Java Framework支援

- ◆ Spring MVC Framework

- ◆ <https://spring.io/>



- ◆ JSF (JavaServer Faces)

- ◆ <https://www.oracle.com/technetwork/k/java/javaee/jaserverfaces-139869.html>



- ◆ Struts

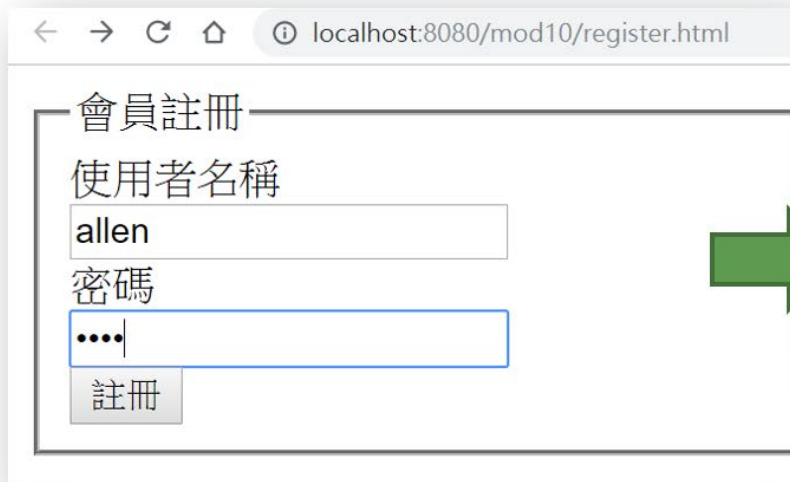
- ◆ <https://struts.apache.org/>

- ◆ 巨匠Java產學課程



Lab

- ◆ 使用MVC架構整合MySQL，設計一個會員註冊作業。



← → ↻ 🏠 ⓘ localhost:8080/mod10/register.html

會員註冊

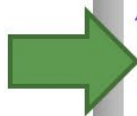
使用者名稱

allen

密碼

....

註冊



← → ↻ 🏠 ⓘ localhost:8080/mod10/RegisterServlet

使用者 allen 完成註冊!!