
Java EE7 Web

網路元件容器工具

鄭安翔

ansel_cheng@hotmail.com

課程大綱

- 1) 部署網路應用程式
 - 網路應用程式組成
 - **Web Archive**
 - 部署描述
- 2) Servlet 初始參數
- 3) 錯誤回報

網路應用程式組成

- 網路元件容器可能包含多個網路應用程式
 - 每個網路應用程式賦予一個唯一的**Context**根節點
 - **Context**代表在瀏覽器中使用的網路應用程式名稱
 - URL中以**Context**名稱作前綴,區分不同應用程式
`http://localhost:8080/Context/index.html`
- 網路應用程式組成
 - 靜態網頁資源
 - 直接置於**Context**目錄之下
 - URL 路徑可以對應到應用程式結構的實體檔案作存取
 - 隱藏子目錄**WEB-INF**
 - 資源無法由URL 路徑直接對應存取
 - 透過部署描述檔或**annotation**標註描述URL與資源對應關係

Web Application : Context

■ Context

一個完整,獨立的網路應用程式單元

□ 靜態資源

■ HTML 檔案

■ JSP 檔案

■ 圖形檔

■ 音效檔

□ 設定資料

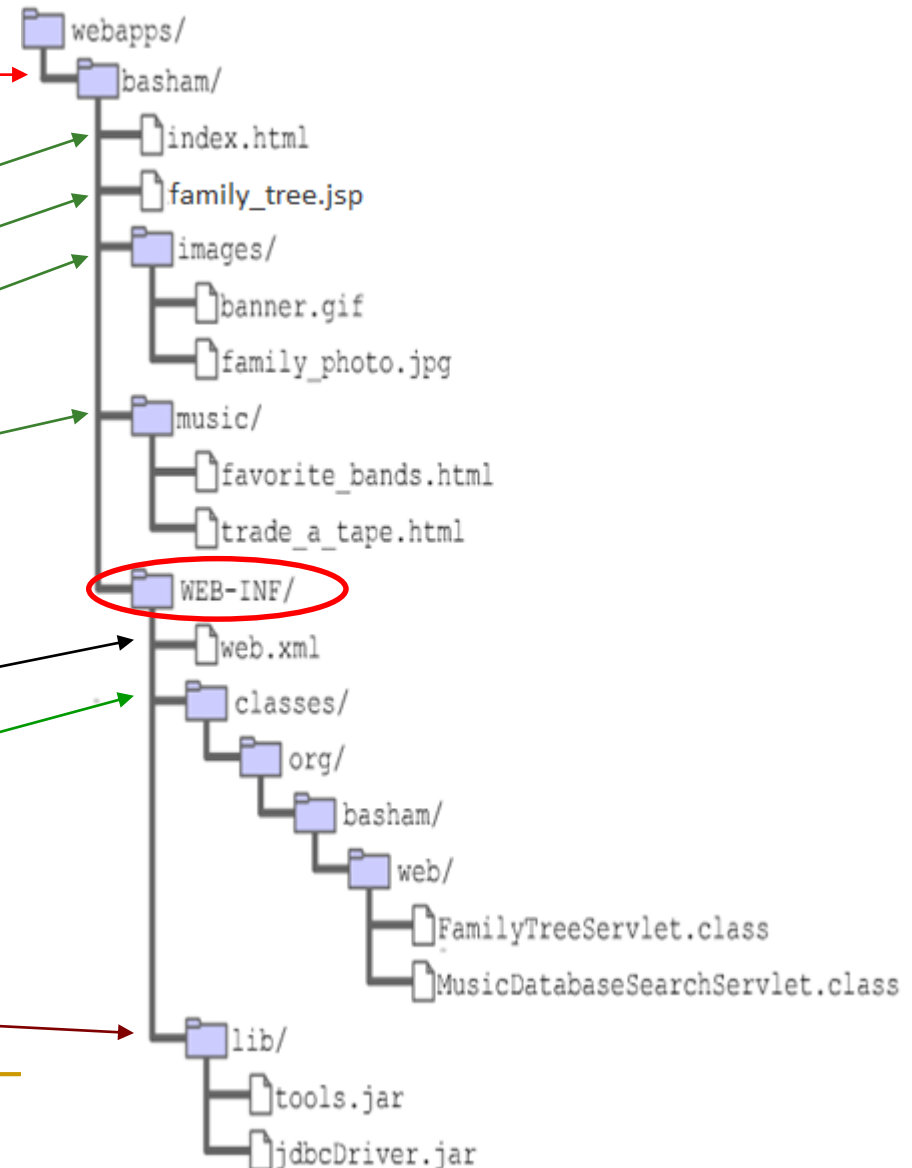
■ Deployment Descriptor

□ 動態程式

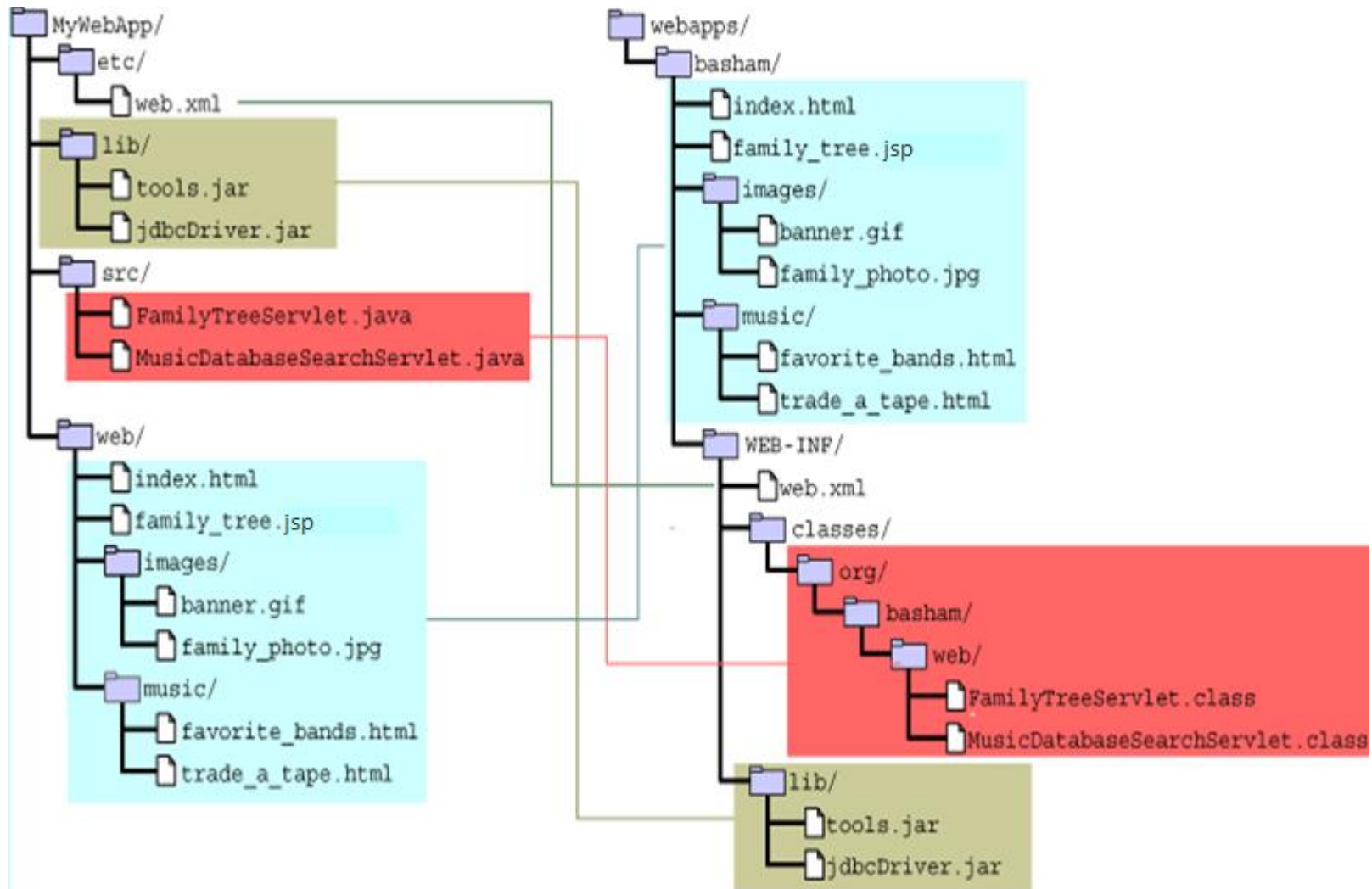
■ Java Servlet

□ 外部資源

■ Java Archive



Development to Deployment Mapping



Web Archive

- 將整個網路應用程式壓縮成一個檔案
 - Java SE 打包格式為JAR歸檔文件
 - zip格式
 - 網路應用程式則使用WAR歸檔文件
 - 符合JavaEE網路應用程式規範標準的JAR格式
 - 副檔名為 .war
 - 預設WAR檔名稱即為Context名稱
 - 內部結構與應用程式部署的結構相同
 - 可直接置於webapps資料夾之下(Tomcat)部署
 - 有利於檔案傳送及部署
 - NetBeans IDE 製作war檔
 - Clean and Build
 - <Project Folder> \ dist 資料夾

部署描述 Deployment Descriptor

■ 部署描述

- 向Servlet Container描述網路應用程式結構,使容器可以正確的尋找並執行應用程式元件
- Java EE 5 之前
 - 部署描述資訊置於<context path>\WEB-INF\web.xml
 - 此檔案無法透過網路存取
- Java EE 6 之後,有三種作法
 - 部署描述檔 web.xml
 - JAR檔中META-INF/web-fragment.xml
 - Jar檔作為網路應用程式的部份模組
 - 原始檔內使用annotation標註
 - 並非所有設定資訊均可透過annotation完成

部署描述控制

■ 部署描述控制

□ XML檔案描述覆蓋標註描述

□ <metadata-complete>標籤

- web.xml中使用：web-fragment.xml或annotation標註無效
- web-fragment.xml中使用：其他web-fragment.xml或annotation標註無效

□ <absolute-ordering>、<ordering>、<name>標籤


- <name>標籤為部署描述命名
- <absolute-ordering>、<ordering>標籤指定設定順序

web.xml常用功能

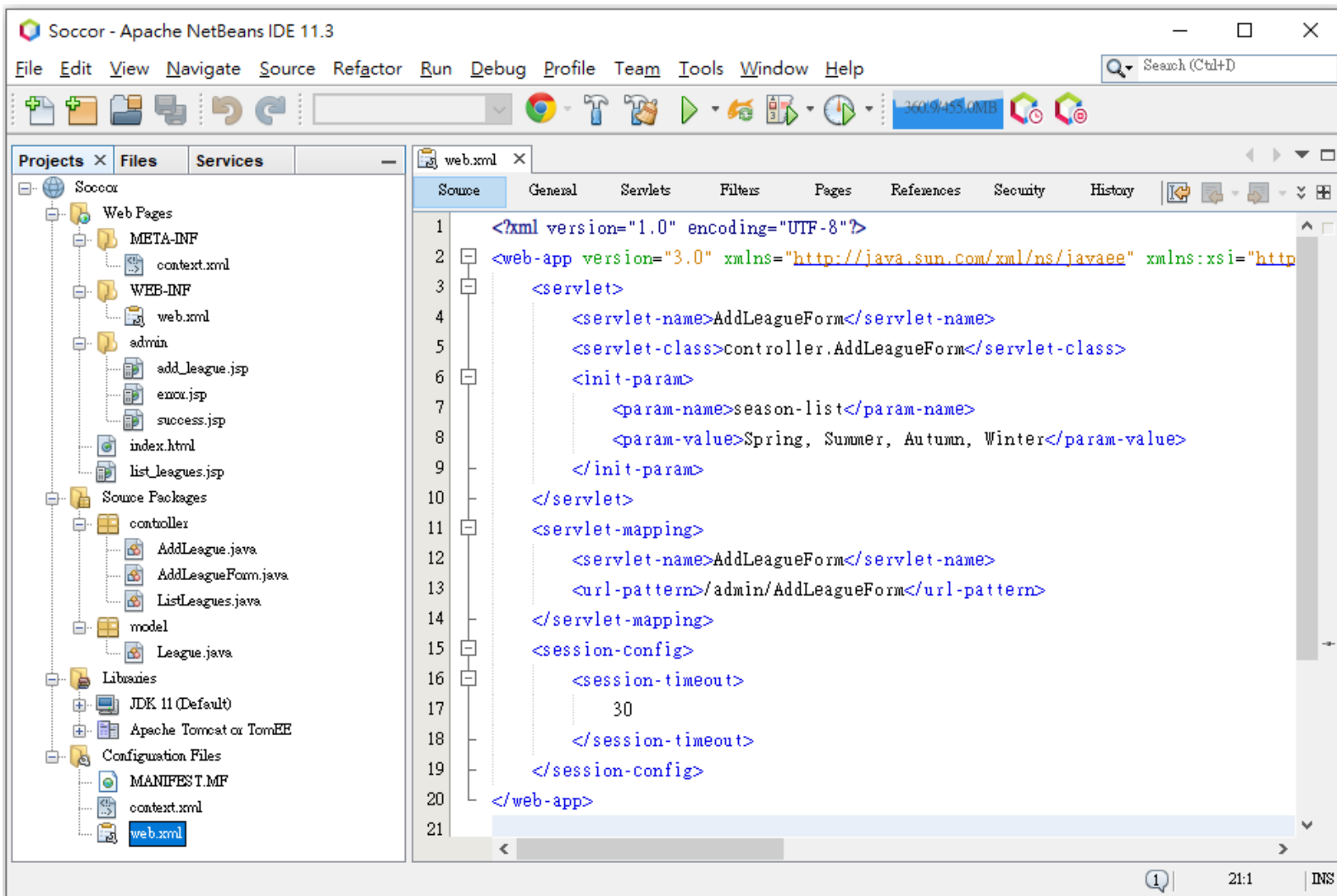
功能	Tag
定義起始參數 (context/Servlet)	<context-param > <init-param>
Servlet定義 Servlet名稱對應	<servlet> <servlet-mapping>
Session管理	<session-config>
錯誤處理機制	<error-page>
安全性控管	<security-constraint> <login-config> <security-role>

Deployment Descriptor – Servlet Mapping

```
<web-app>
  <display-name> Web application name. </display-name>
  <description> Web application description. </description>
  <servlet>
    <servlet-name> Servlet 名稱. </servlet-name>
    <servlet-class> Servlet 類別全名 (Fully Qualified Name). </servlet-class>
    <init-param>
      <param-name> Servlet parameter name. </param-name>
      <param-value> Servlet parameter name's value. </param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name> Servlet 名稱. </servlet-name>
    <url-pattern> Servlet URL 對應. </url-pattern>
  </servlet-mapping>
</web-app>
```



NetBeans 管理 web.xml



NetBeans 管理 web.xml

The image displays four screenshots of the NetBeans IDE, illustrating the configuration of the `web.xml` file through different tabs in the configuration window.

Top Left Screenshot (General Tab): The `web.xml` file is open, and the **General** tab is selected. The **Context Parameters** section shows a table with the following data:

Parameter Name	Parameter Value
leaguesFile	/WEB-INF/data/leagues.txt
season-list	Spring, Summer, Fall, Winter

Buttons for **Add...**, **Edit...**, and **Remove** are visible below the table. The **Web Application Listeners** section is also visible.

Top Right Screenshot (Pages Tab): The `web.xml` file is open, and the **Pages** tab is selected. The **Welcome Files** and **Error Pages** sections are visible. The **JSP Property Groups** section has an **Add JSP Property Group...** button.

Bottom Left Screenshot (Filters Tab): The `web.xml` file is open, and the **Filters** tab is selected. The **Servlet Filters** and **Filter Mappings** sections are visible.

Bottom Right Screenshot (References Tab): The `web.xml` file is open, and the **References** tab is selected. The **Environment Entries**, **Resource References**, **Resource Environment References**, **EJB References**, and **Message Destination References** sections are visible.

Security Tab (Bottom Right Screenshot): The `web.xml` file is open, and the **Security** tab is selected. The **Login Configuration**, **Security Roles**, and **Security Constraints** sections are visible. The **Security Constraints** section has an **Add Security Constraint** button.

URL 多重對應及萬用字元

```
<servlet-mapping>  
    <servlet-name>MyServlet</servlet-name>  
    <url-pattern>/MyServlet</url-pattern>  
    <url-pattern>/YourServlet</url-pattern>  
    <url-pattern>/HisServlet/*</url-pattern>  
</servlet-mapping>
```

http://localhost:8080/MyAppRoot/MyServlet

http://localhost:8080/MyAppRoot/YourServlet

http://localhost:8080/MyAppRoot/HisServlet

http://localhost:8080/MyAppRoot/HisServlet/one

http://localhost:8080/MyAppRoot/HisServlet/twenty

使用 Annotation 標註

■ Annotation 標註描述作部署描述

```
import javax.servlet.annotation.WebServlet  
  
.....  
@WebServlet(name="MyServletName",  
    urlPatterns={"/MyServlet", "/YourServlet", "/HisServlet/*" })  
public class MyServlet {  
    .....  
    .....  
}
```

使用Servlet 名稱作頁面轉送

```
<<interface>>  
javax.servlet.ServletContext  
  
getRequestDispatcher(path : String):RequestDispatcher  
getNamedDispatcher(name : String) : RequestDispatcher
```

```
<<interface>>  
javax.servlet.RequestDispatcher  
  
forward( req : ServletRequest, res ServletResponse)  
include (req : ServletRequest, res ServletResponse)
```

```
ServletContext sc = this.getServletContext();  
RequestDispatcher rd = sc.getNamedDispatcher("MyServlet");  
rd.forward(request, response);
```

課程大綱

- 1) 部署網路應用程式
- 2) **Servlet 初始參數**
 - ▣ **Servlet 的生命週期**
 - ▣ **Servlet 初始參數設定**
- 3) 錯誤回報

Servlet Life Cycle 生命週期

■ Servlet 生命週期

- 由Servlet Container容器掌控,開發者無法控制
- 容器建立Servlet時,呼叫其預設(無參數)建構子
 - Servlet 沒有包含傳入參數的建構子
 - Servlet 屬性初始化???
- Servlet 介面定義生命週期方法
 - init() : 屬性初始化
 - destroy() : 屬性儲存

生命週期方法

- Servlet interface提供三個生命週期相關方法

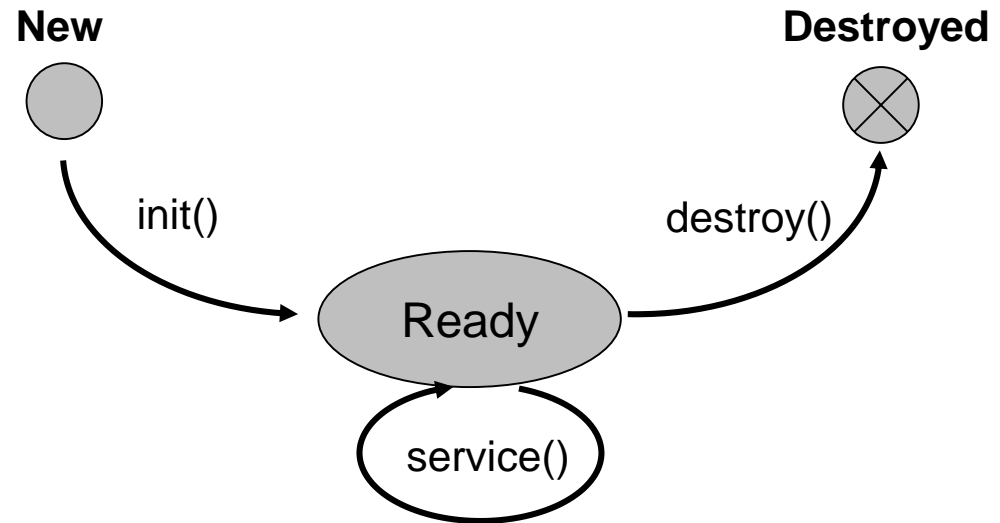
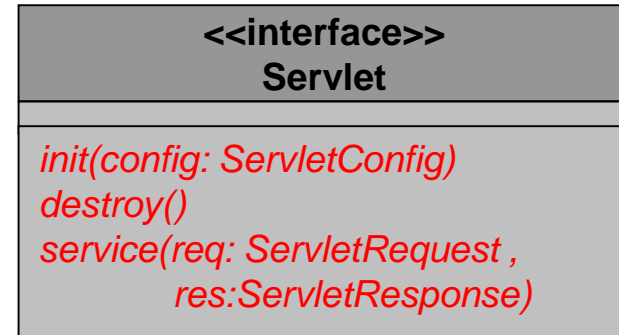
<<interface>> Servlet
<i>getServletConfig() : ServletConfig</i> <i>getServletInfo() : String</i> <i>init(config: ServletConfig)</i> <i>destroy()</i> <i>service(req: ServletRequest , res:ServletResponse)</i>

生命週期 方法	執行時機	用途
init()	Servlet 被建立時,容器呼叫init方法進行初始化, 每個Servlet類別只呼叫一次	設定初始參數 建立Servlet instance所需資源
service()	User 發出 HTTP 請求, 容器會呼叫service來提供服務	根據HTTP請求方法,呼叫對應的doXXX()
destroy()	容器判斷某個Servlet實體已不再需要時呼叫,釋放該Servlet佔用的資源	儲存Servlet 狀態 釋放Servlet instance佔用的資源

Servlet Life Cycle 生命週期

■ Servlet生命週期

- 建立Servlet 物件
 - 載入Servlet 類別
 - 建立Servlet 實體
 - 呼叫init()方法
- 提供服務
 - 呼叫service()方法
- 回收Servlet物件
 - 呼叫 destroy()方法

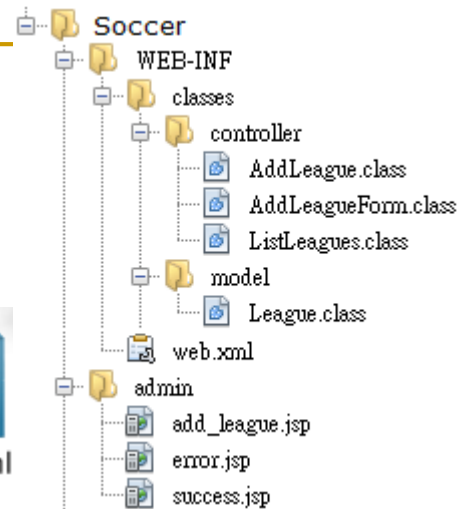


載入Servlet類別

1. 辨識對應Servlet元件
2. 載入Servlet類別



web.xml



```
web.xml
Source  General  Servlets  Filters  Pages  References  Security  History
1  <?xml version="1.0" encoding="UTF-8"?>
2  <web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://
3  <servlet>
4      <servlet-name>AddLeagueForm</servlet-name>
5      <servlet-class>controller.AddLeagueForm</servlet-class>
6      <init-param>
7          <param-name>season-list</param-name>
8          <param-value>Spring, Summer, Autumn, Winter</param-value>
9      </init-param>
10 </servlet>
11 <servlet-mapping>
12     <servlet-name>AddLeagueForm</servlet-name>
13     <url-pattern>/admin/AddLeagueForm</url-pattern>
14 </servlet-mapping>
15 <session-config>
20 </web-app>
```

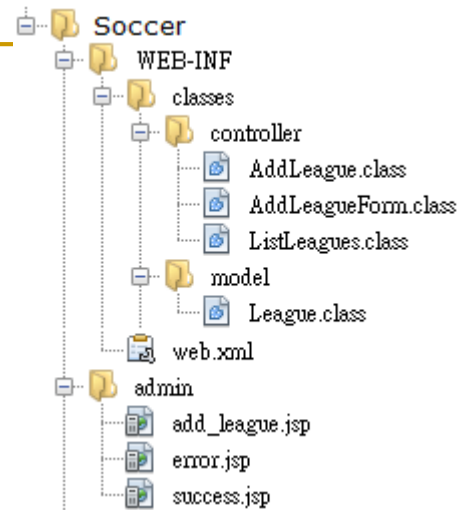
網路伺服器

Web Container

<http://localhost:8080/admin/AddLeagueForm>

建立Servlet 物件

3. 建立Servlet實體
4. 呼叫 `init()` 初始化Servlet



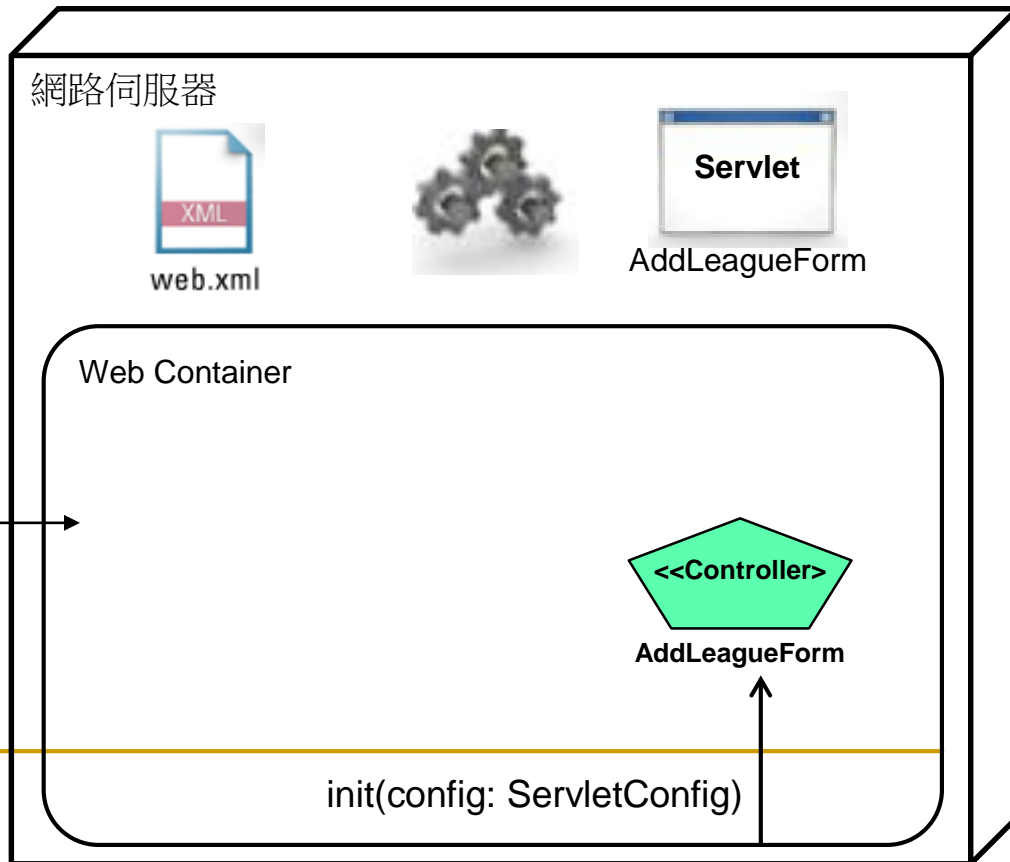
web.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://
3   <servlet>
4     <servlet-name>AddLeagueForm</servlet-name>
5     <servlet-class>controller.AddLeagueForm</servlet-class>
6     <init-param>
7       <param-name>season-list</param-name>
8       <param-value>Spring, Summer, Autumn, Winter</param-value>
9     </init-param>
10  </servlet>
11  <servlet-mapping>
12    <servlet-name>AddLeagueFo
13    <url-pattern>/admin/AddLe
14  </servlet-mapping>
15  <session-config>
20 </web-app>
```

ServletConfig

season-list=Spring, Summer, Autumn, Winter
--

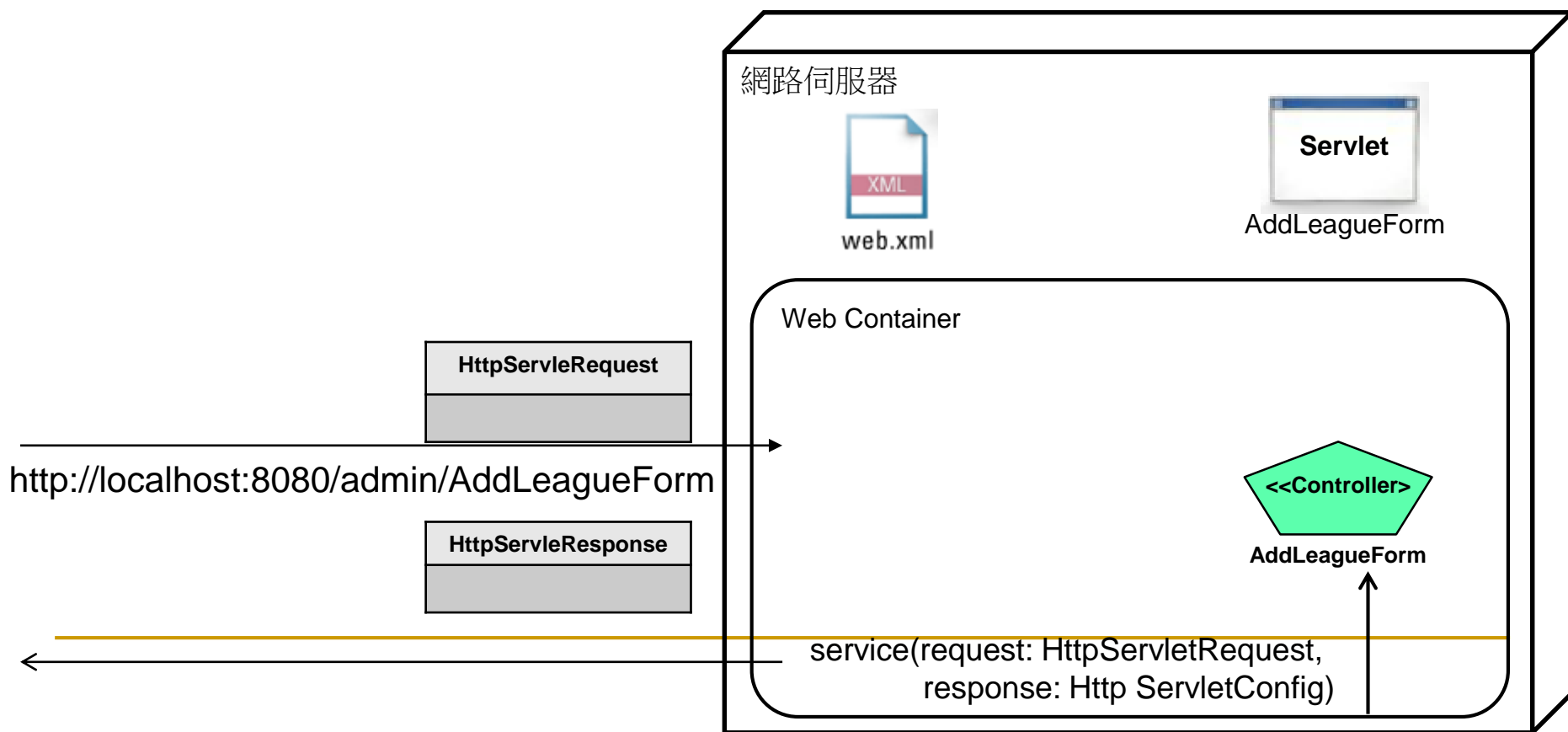
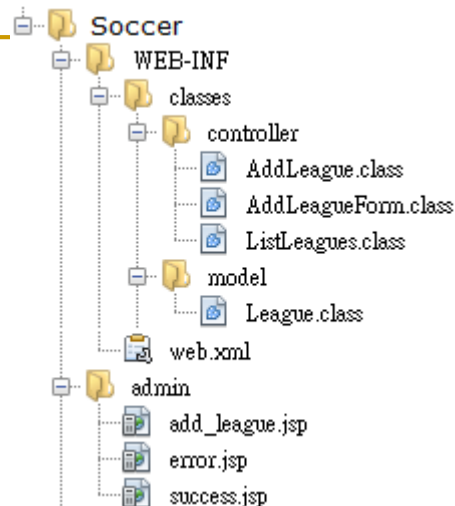
<http://localhost:8080/admin/AddLeagueForm>



提供服務

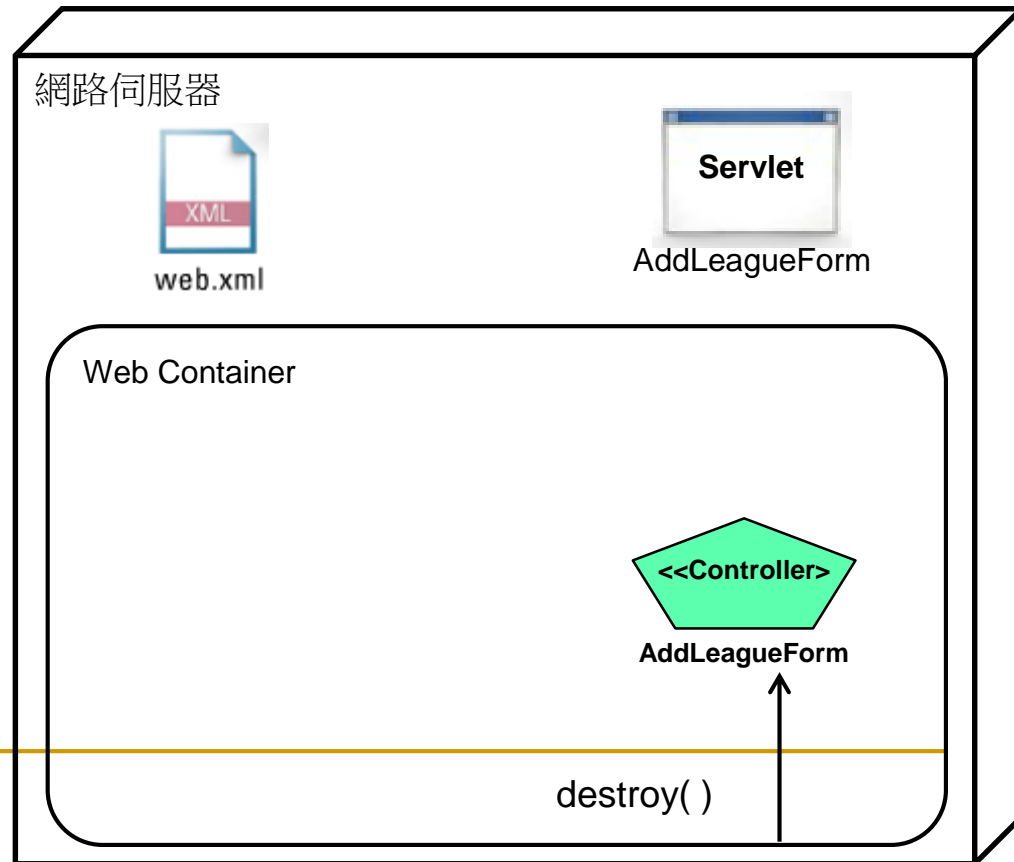
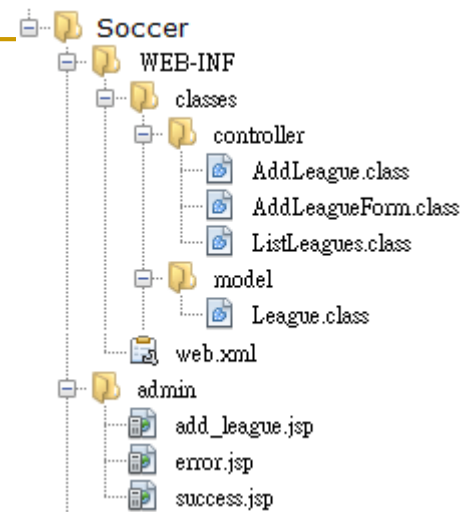
5. 呼叫service()方法

- 建立HttpServletRequest及HttpServletResponse物件



回收Servlet物件

5. 呼叫destroy()方法



部署描述檔 – Servlet init parameter

```
<web-app>
  <display-name> Web application 名稱. </display-name>
  <description> Web application 說明. </description>
  <servlet>
    <servlet-name> Servlet 名稱. </servlet-name>
    <servlet-class> Servlet 類別全名 (Fully Qualified Name). </servlet-class>
    <init-param>
      <param-name> Servlet 初始參數名稱. </param-name>
      <param-value> Servlet 初始參數值. </param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name> Servlet 名稱. </servlet-name>
    <url-pattern> Servlet URL 對應. </url-pattern>
  </servlet-mapping>
</web-app>
```

包裝為
ServletConfig
物件

Annotation 標註- Servlet 初始參數

- Annotation 標註描述作部署描述
 - 類別定義在javax.servlet.annotation套件中
 - javax.servlet.annotation.WebServlet
 - javax.servlet.annotation.WebInitParam

```
import javax.servlet.annotation.WebServlet;
import javax.servlet.annotation.WebInitParam;

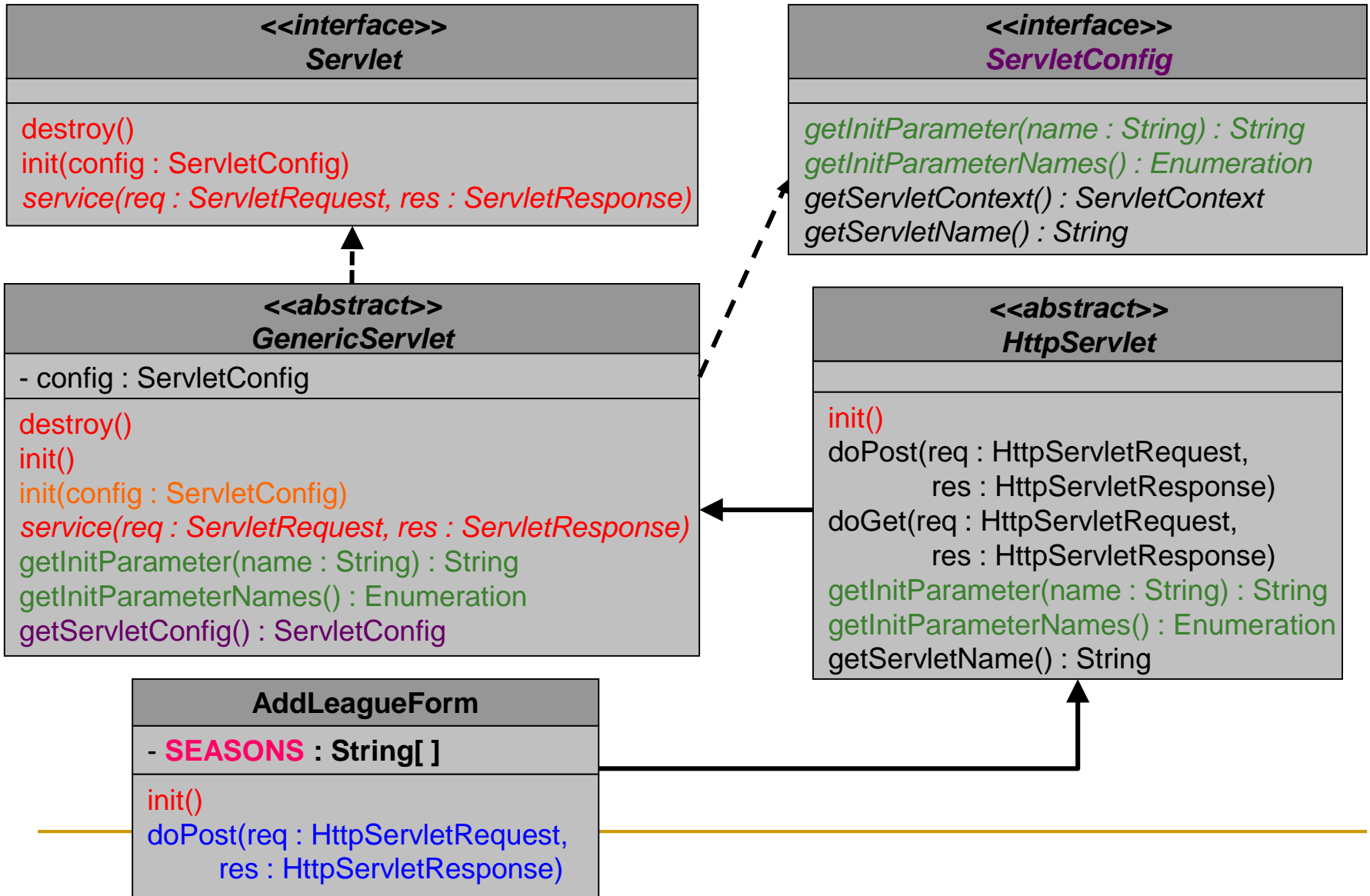
.....

@WebServlet(name="MyServletName", urlPatterns={"/MyServlet"},
initParams={@WebInitParam(name="paramname", value="paramvalue")})
public class MyServlet {

    .....

    .....
}
```

HttpServlet – Lift Cycle Methods



Soccer 範例



增加新聯盟



<http://localhost:8080/Soccer/index.html>

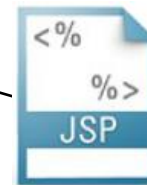


Index.html

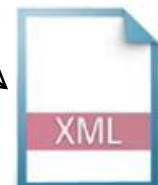
<http://localhost:8080/Soccer/admin/AddLeagueForm>



AddLeagueForm

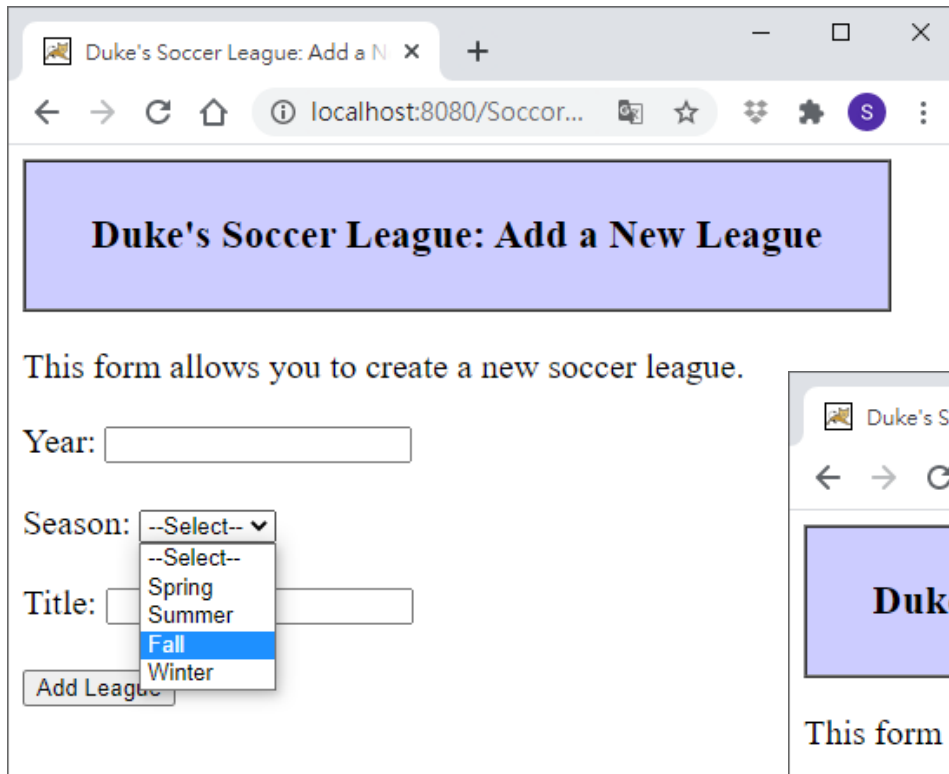


add_league.jsp



web.xml

Soccer 範例



Duke's Soccer League: Add a New League

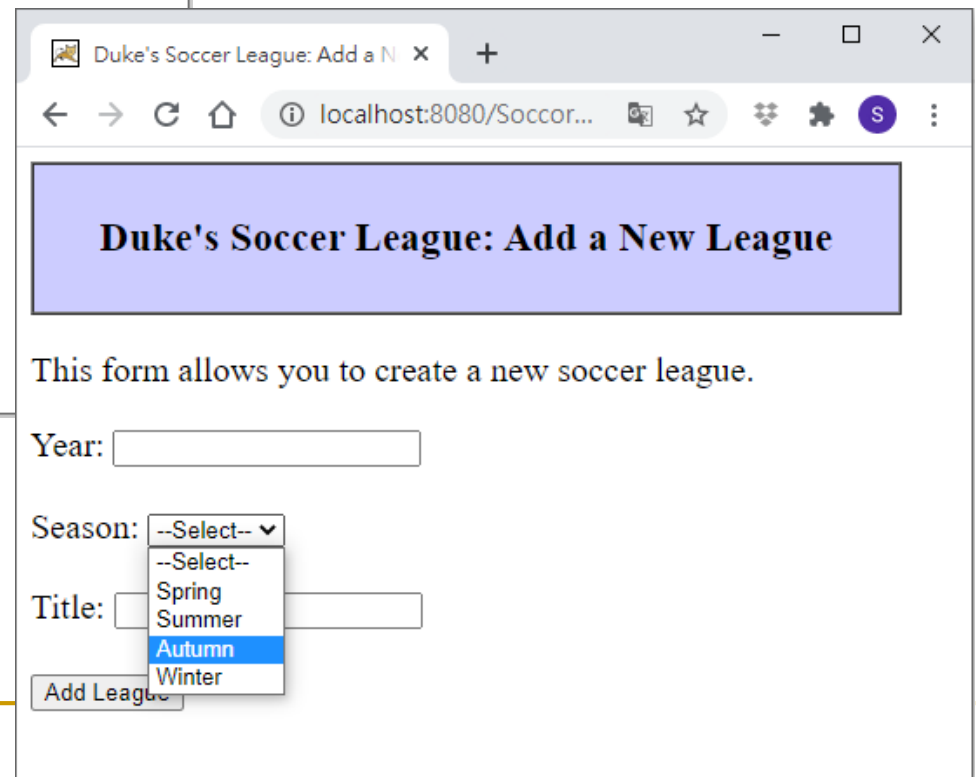
This form allows you to create a new soccer league.

Year:

Season:

Title:

Add League



Duke's Soccer League: Add a New League

This form allows you to create a new soccer league.

Year:

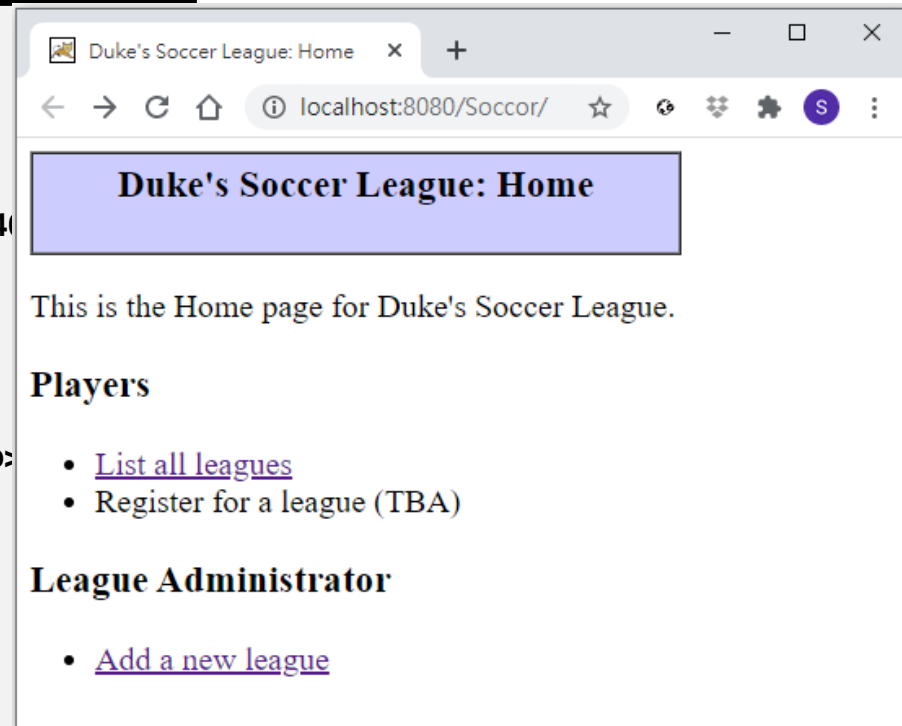
Season:

Title:

Add League

Soccer 範例

```
<html>
<head>
  <title>Duke's Soccer League: Home</title>
</head>
<body>
  <table border='1' cellpadding='5' cellspacing='0' width='400px'>
    <tr bgcolor='#CCCCFF' align='center'
      valign='center' height='20'>
      <td><h3>Duke's Soccer League: Home</h3></td>
    </tr>
  </table>
  <p>This is the Home page for Duke's Soccer League. </p>
  <h3>Players</h3>
  <ul>
    <li><a href='ListLeagues'>List all leagues</a></li>
    <li>Register for a league (TBA)</li>
  </ul>
  <h3>League Administrator</h3>
  <ul>
    <li><a href='admin/AddLeagueForm'>Add a new league</a></li>
  </ul>
</body>
</html>
```



Soccer 範例

```
package controller;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;
import javax.servlet.annotation.WebInitParam;

@WebServlet(name = "AddLeagueForm", urlPatterns = {"/admin/AddLeagueForm"},
    initParams={@WebInitParam(name="season-list", value="Spring, Summer, Fall, Winter")})
public class AddLeagueForm extends HttpServlet {
    private String[] SEASONS;
    public void init() throws ServletException {
        String seasonStr = this.getInitParameter("season-list");
        SEASONS = seasonStr.split(", ");
    }
    protected void processRequest(HttpServletRequest request, HttpServletResponse
        response) throws ServletException, IOException {
        RequestDispatcher rd = request.getRequestDispatcher("admin/add_league.jsp");
        request.setAttribute("seasonList", SEASONS);
        rd.forward(request, response);
    }
    .....
}
```

Soccer 範例

web.xml

Source General **Servlets** Filters Pages References Security History Add...

Servlets Add Servlet Element...

AddLeagueForm -> /admin/AddLeagueForm Remove

Servlet Name: AddLeagueForm Startup Order:

Description:

☒ Servlet Class: controller.AddLeagueForm Browse... [Go to Source](#)

☐ JSP File: Browse... [Go to Source](#)

URL Pattern(s): /admin/AddLeagueForm
Use comma (,) to separate multiple patterns.

Initialization Parameters:

Parameter Name	Parameter Value	Description
season-list	Spring, Summer, Fall, Winter	

Add... Edit... Remove

Security Role References:

Role Ref Name	Role Ref Link	Description
---------------	---------------	-------------

Add... Edit... Remove

Run As:

Add Initialization Parameter

Parameter Name: season-list

Parameter Value: Spring, Summer, Fall, Winter

Description:

OK Cancel

web.xml

Source General **Servlets** Filters Pages References Security History Add...

Servlets Add Servlet Element...

AddLeagueForm -> /admin/AddLeagueForm Remove

Servlet Name: AddLeagueForm Startup Order:

Description:

☒ Servlet Class: controller.AddLeagueForm Browse... [Go to Source](#)

☐ JSP File: Browse... [Go to Source](#)

URL Pattern(s): /admin/AddLeagueForm
Use comma (,) to separate multiple patterns.

Initialization Parameters:

Parameter Name	Parameter Value	Description
season-list	Spring, Summer, Fall, Winter	

Add... Edit... Remove

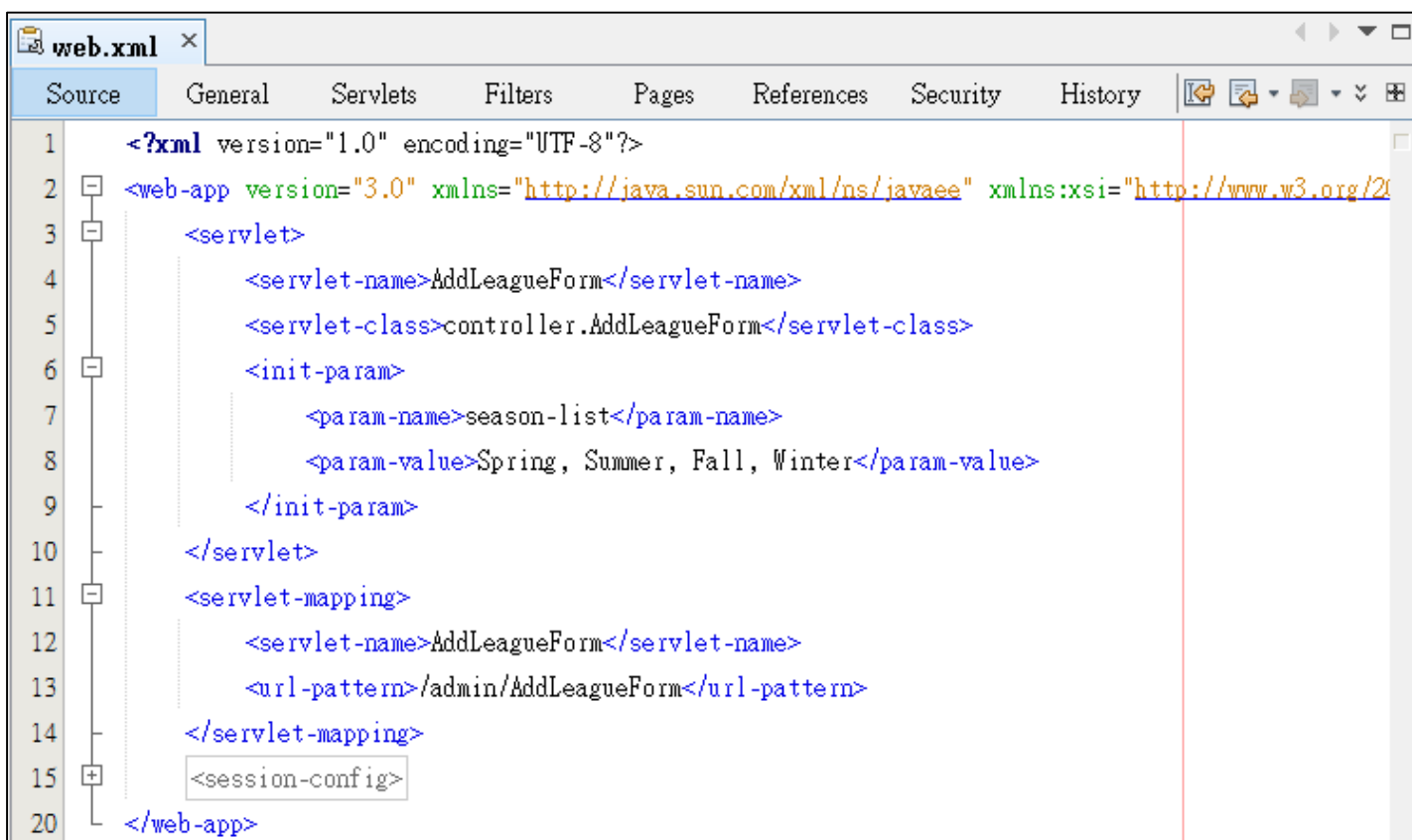
Security Role References:

Role Ref Name	Role Ref Link	Description
---------------	---------------	-------------

Add... Edit... Remove

Run As:

Soccer 範例



```
web.xml x
Source General Servlets Filters Pages References Security History
1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app version="3.0" xmlns="http://java.sun.com/xml/ns/javaee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-app_3_0.xsd">
3   <servlet>
4     <servlet-name>AddLeagueForm</servlet-name>
5     <servlet-class>controller.AddLeagueForm</servlet-class>
6     <init-param>
7       <param-name>season-list</param-name>
8       <param-value>Spring, Summer, Fall, Winter</param-value>
9     </init-param>
10  </servlet>
11  <servlet-mapping>
12    <servlet-name>AddLeagueForm</servlet-name>
13    <url-pattern>/admin/AddLeagueForm</url-pattern>
14  </servlet-mapping>
15  <session-config>
16    <session-timeout>30</session-timeout>
17  </session-config>
18 </web-app>
```



```

<%@ page contentType="text/html"%>
<%@ page session="false" %>

<html>
  <head>
    <title>Duke's Soccer League: Add a New League</title>
  </head>
  <body bgcolor='white'>
    <!-- Page Heading -->
    <table border='1' cellpadding='5' cellspacing='0' width='4
      <tr bgcolor='#CCCCFF' align='center' valign='center' h
        <td><h3>Duke's Soccer League: Add a New League
      </td>
    </tr>
  </table>
  <p>This form allows you to create a new soccer league.<
  <form action='AddLeague' method='POST'>
    Year: <input type='text' name='year' /> <br/><br/>
    Season: <select name='season'>
      <option value='UNKNOWN'> --Select-- </option>
    <%
      String[] SEASONS = (String[])(request.getAttribute("seasonList"));
      for ( int i = 0; i < SEASONS.length; i++ ) {
    <%
      <option value='<%= SEASONS[i] %>'> <%= SEASONS[i] %> </option>
    <%
      }
    <%
      </select> <br/><br/>
      Title: <input type='text' name='title' /> <br/><br/>
      <input type='Submit' value='Add League' />
    </form>
  </body>
</html>

```

Duke's Soccer League: Add a New League

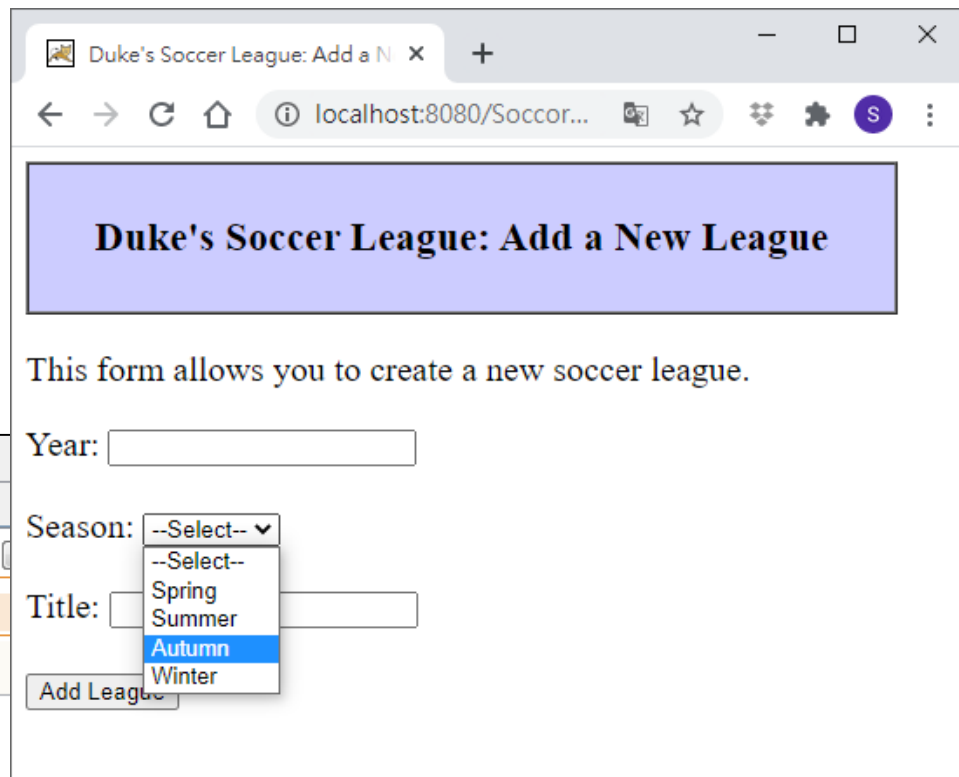
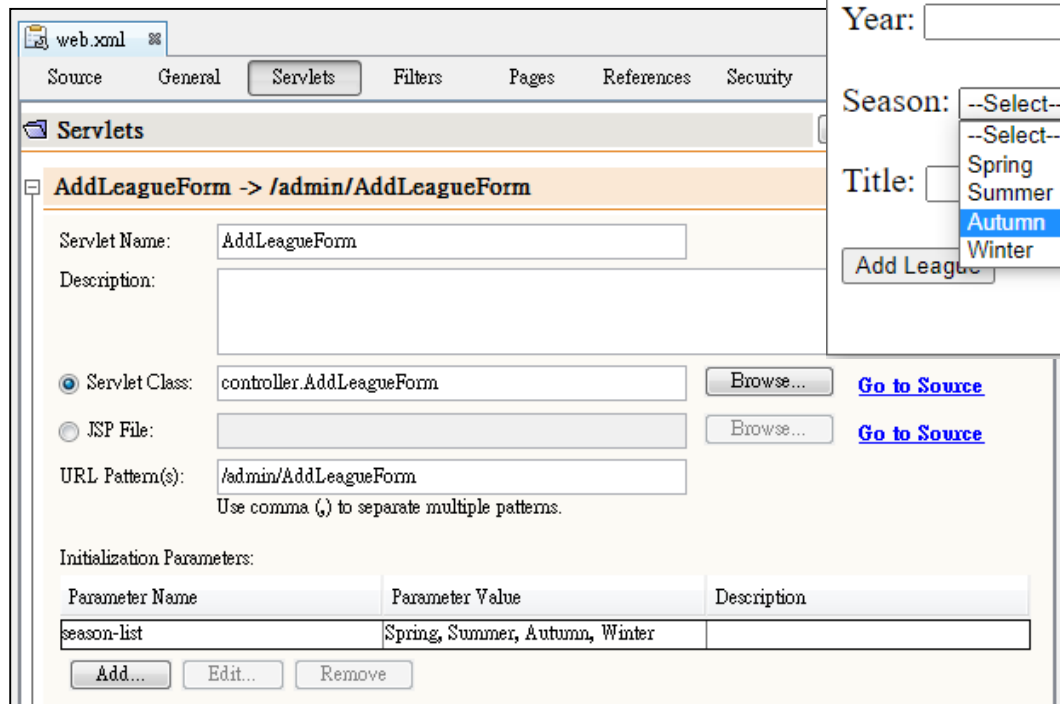
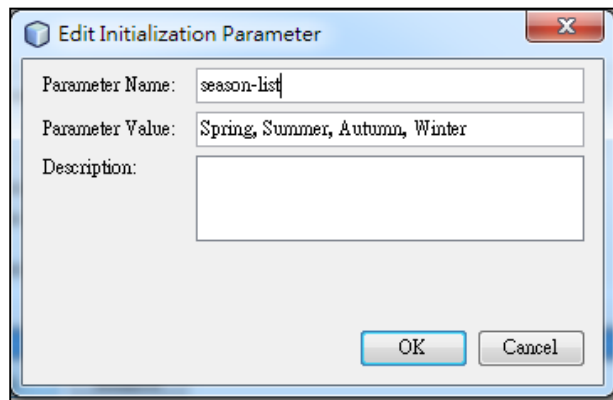
This form allows you to create a new soccer league.

Year:

Season:
 --Select--
 --Select--
 Spring
 Summer
 Fall
 Winter

Title:

Soccer 範例



Lab DVDLibrary 應用程式設計



DVD Library 網路應用程式

列出 DVD 清單

新增 DVD 加入清單

Lab 1- 動態表單



Library User

新增 DVD 加入清單

<http://localhost:8080/DVDLibrary/index.html>

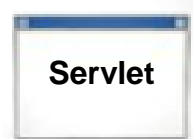


index.html

<http://localhost:8080/DVDLibrary/AddDVDForm>



Library User



AddDVDFormServlet



addDVD.jsp



web.xml

Lab 1

- 使用NetBeans IDE，建立DVDLibrary 網路應用專案
- 編輯 index.html
 - 新增一個連結,連結至AddDVDForm
 - 畫面如下



Lab 1

- 撰寫 `controller.AddDVDServlet.java`
 - Servlet URL 為 `AddDVDForm`
 - 宣告私有屬性 `String[] genres`
 - 覆寫 `init()` 方法
 - 取得Servlet初始參數 `genres-list`
 - 使用字串切割方法 `split()`, 將傳回之字串陣列存在 `genres` 屬性
 - `processRequest()` 方法
 - 將 `genres` 屬性, 加入請求屬性 `genreList`
 - 將請求轉送至 `add_dvd.jsp`

Lab 1

- 撰寫add_dvd.jsp
 - 建立一個Form表單如右
 - 送出Post方法至Add_DVD
 - Text 輸入欄位 title, year,
 - Select 欄位 genre
 - 取得請求屬性genreList
 - 轉型為字串陣列
 - 陣列中字串作為Option
 - Text 輸入欄位otherGenre
 - Submit 按鈕 save
- 測試、執行

Add DVD Form

localhost:8080/DVDLibrary...

新增 DVD

片名:

年份:

類型: 或其他種類:

- Animation
- Action
- Sci-Fi

課程大綱

- 1) 部署網路應用程式
- 2) **Servlet** 初始參數
- 3) **錯誤回報**
 - 日誌檔
 - 錯誤頁面

訊息寫入日誌檔 log file

<<interface>>
ServletConfig

getInitParameter(name : String) : String
getInitParameterNames() : Enumeration
getServletContext() : ServletContext
getServletName() : String

<<abstract>>
GenericServlet

- config : ServletConfig

destroy()
getInitParameter(name : String) : String
getInitParameterNames(name : String) : String
getServletConfig() : ServletConfig
getServletContext() : ServletContext
getServletInfo() : String
getServletName() : String
init()
init(config : ServletConfig)
log(msg : String)
log(msg : String, excp : Throwable)
service(ServletRequest req, ServletResponse res)

<<interface>>
ServletContext

setAttribute(name : String, object : Object)
getAttribute(name : String) : Object
getAttributeNames() : Enumeration
removeAttribute(name : String)
getContext(uripath : String) : Object
getMajorVersion() : int
getMimeType() : int
getNameDispatcher(name : String) : RequestDispatcher
getInitParameter(name : String) : String
getInitParameterNames() : Enumeration
getResource(path : String) : URL
getResourceAsStream(path : String) : InputStream
getResourcePaths() : java.util.Set
getServletInfo() : String
getServletContextName() : String
log(msg : String)
log(msg : String, excp Throwable)

例外處理原則

- **Servlet 或JSP 程式執行時發生例外處理方式**
 - 自訂錯誤頁面描述問題
 - 顯示使用者易懂的訊息
 - 保障系統安全性
 - 直接將例外丟出
 - 盡量避免直接向用戶丟出**Exception**例外
 - 被包含的網頁片段可拋出例外, 以交由最適處理例外的位置(主網頁)處理例外
- **包裹(wrap)發生的例外**
 - **ServletException**：持續性的請求失敗
 - **UnavailableException**：暫時性的請求失敗

JSP 錯誤頁面組態資訊

■ JSP 指令標籤 Directive Tag

- Page 指令標籤, 設定JSP頁面組態資訊

■ errorPage

- 當例外發生時,所導向之錯誤頁面

`<%@ page errorPage="error.jsp" %>`

■ isErrorPage

- 頁面是否為錯誤頁面
- 包含 exception 隱含變數

`<%@ page isErrorPage="true" %>`

隱含變數	型別	用途	Scope
exception	java.lang.Throwable	其他JSP所拋出之例外事件, 只在錯誤頁面中存在	Page

部署描述檔 – 例外頁面

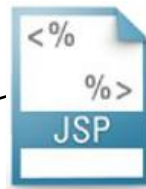
```
<web-app>
  <display-name> Web application 名稱. </display-name>
  <description> Web application 說明. </description>
  <servlet> .... </servlet>
  <servlet-mapping> .... </servlet-mapping>
  <context-param> .... </context-param>

  <error-page>
    <error-code> 404 </error-code>
    <location> /404.jsp </location>
  </error-page>
  <error-page>
    <exception-type> java.lang.NullPointerException </exception-type>
    <location> /error.jsp </location>
  </error-page>
</web-app>
```

Soccer 範例



增加新聯盟



add_league.jsp



AddLeagueForm

<http://localhost:8080/Soccer/admin/AddLeagueForm>

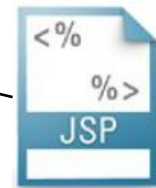


AddLeague

<http://localhost:8080/Soccer/admin/AddLeague>



League

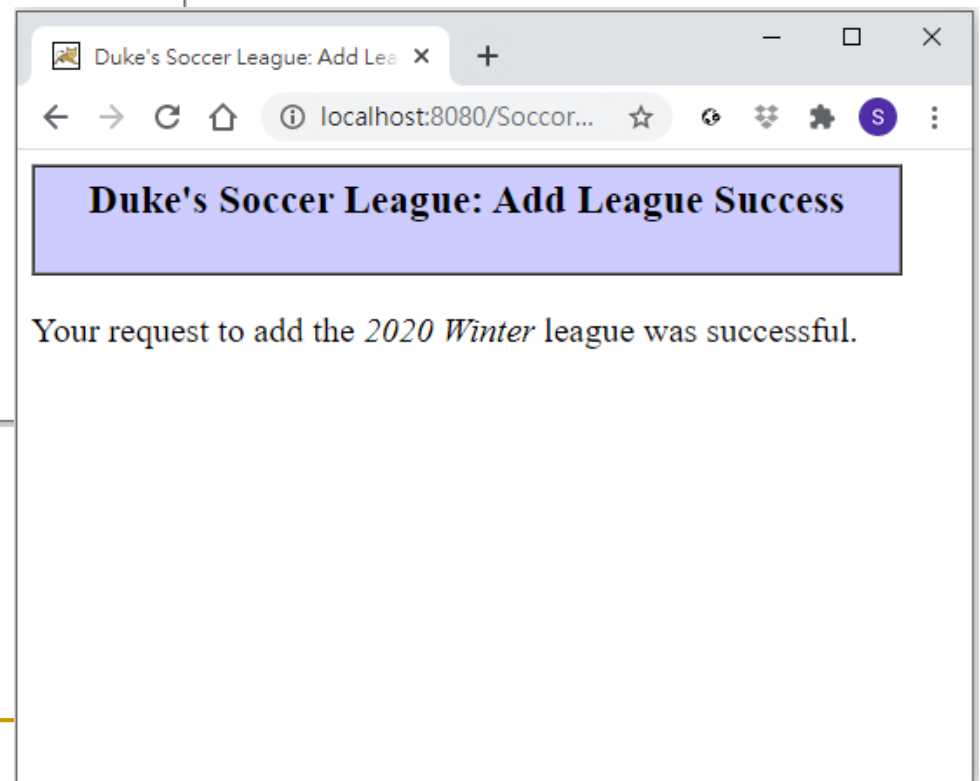
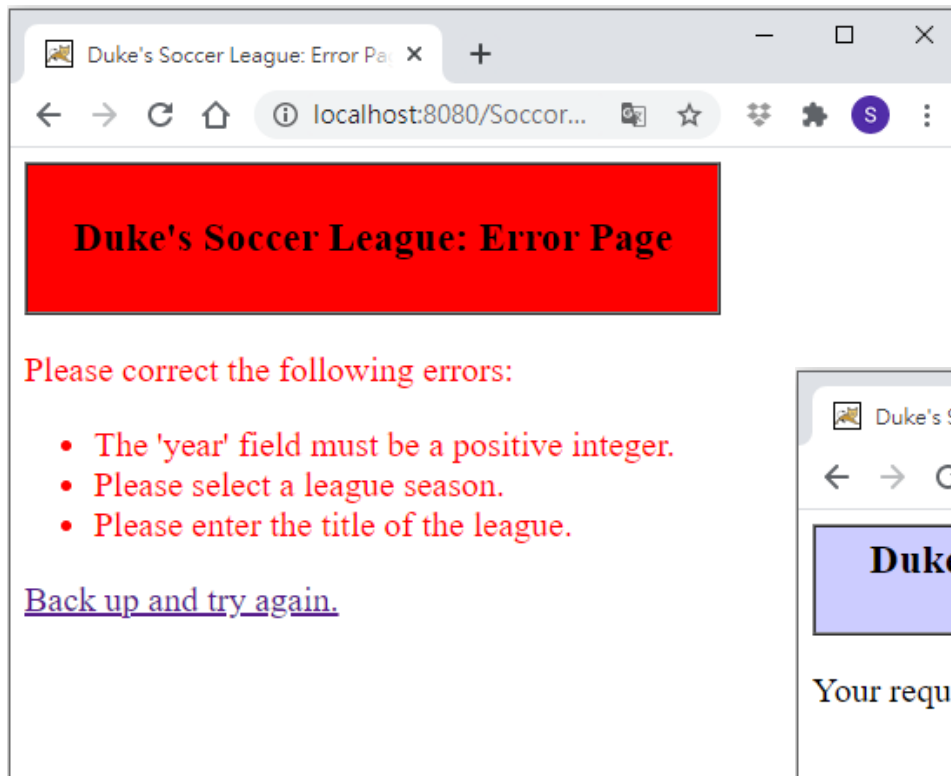


error.jsp



success.jsp

Soccer 範例



Soccer 範例

```
package controller;
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import javax.servlet.annotation.WebServlet;
import model.*;

@WebServlet(name = "AddLeague", urlPatterns = {"/admin/AddLeague"})
public class AddLeague extends HttpServlet {

    protected void processRequest(HttpServletRequest request, HttpServletResponse
                                response) throws ServletException, IOException {
        // Keep a set of strings to record form processing errors.
        List<String> errorMsgs = new LinkedList<String>();
        // Store this set in the request scope, in case we need to send the ErrorPage.
        request.setAttribute("errorMsgs", errorMsgs);
        try {
            // Retrieve form parameters.
            String yearStr = request.getParameter("year").trim();
            String season = request.getParameter("season").trim();
            String title = request.getParameter("title").trim();
            int year = -1;
            try {
                year = Integer.parseInt(yearStr);
            } catch (NumberFormatException nfe) {
                errorMsgs.add("The 'year' field must be a positive integer.");
            }
        }
```

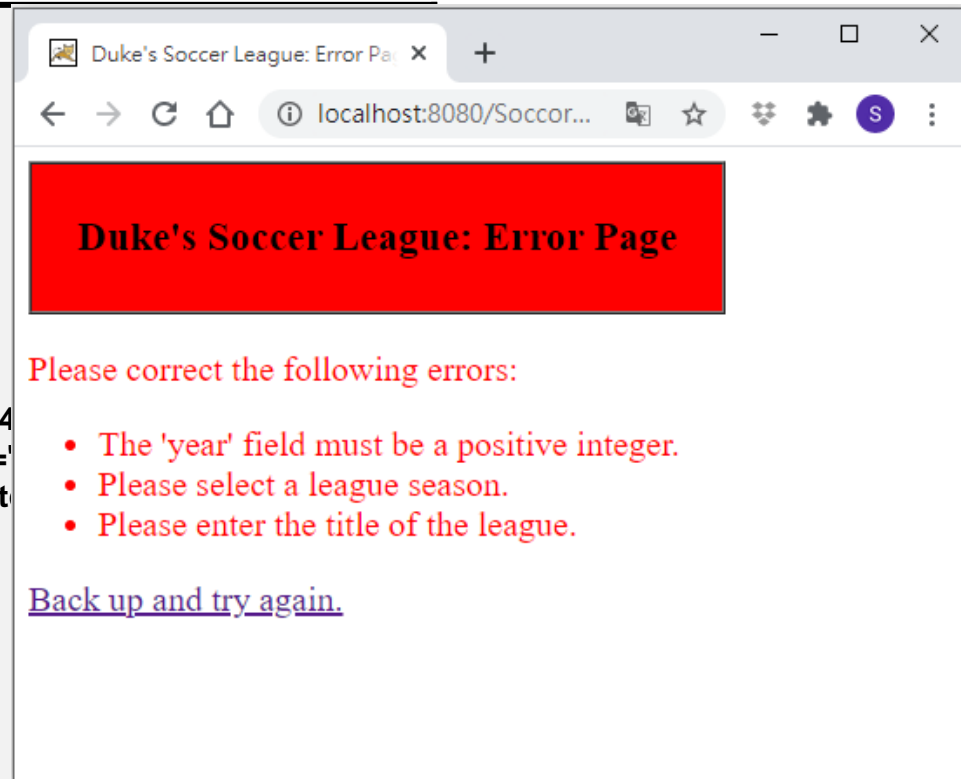
```
// Verify form parameters
if ( (year != -1) && ((year < 2010) || (year > 2025)) ) {
    errorMsgs.add("The 'year' field must within 2010 to 2025.");
}
if ( season.equals("UNKNOWN") ) {
    errorMsgs.add("Please select a league season.");
}
if ( title.length() == 0 ) {
    errorMsgs.add("Please enter the title of the league.");
}
// Send the ErrorPage view if there were errors
if ( ! errorMsgs.isEmpty() ) {
    RequestDispatcher view = request.getRequestDispatcher("admin/error.jsp");
    view.forward(request, response);
    return;
}
// Perform business logic
League league = new League(year, season, title);
// Store the new league in the request-scope
request.setAttribute("league", league);
// Send the Success view
RequestDispatcher view = request.getRequestDispatcher("admin/success.jsp");
view.forward(request, response);
} catch (RuntimeException e) {
    // Log Exception Message
    this.log(e.toString());
    errorMsgs.add(e.getMessage());
    RequestDispatcher view = request.getRequestDispatcher("admin/error.jsp");
    view.forward(request, response);
} // END of try-catch block
}
}
```



```

<%@ page contentType="text/html"%>
<%@ page session="false" %>
<%@ page import="java.util.*" %>
<%@ page isErrorPage="true" %>
<html>
  <head>
    <title>Duke's Soccer League: Error Page</title>
  </head>
  <body bgcolor='white'>
    <!-- Page Heading -->
    <table border='1' cellpadding='5' cellspacing='0' width='400px'>
      <tr bgcolor='red' align='center' valign='center' height='40px'>
        <td><h3>Duke's Soccer League: Error Page</h3></td>
      </tr>
    </table>
    <%-- Generate main body --%>
    <p>
      <font color='red'>Please correct the following errors:
      <ul>
        <%
          List<String> errorMsgs = (List<String>) request.getAttribute("errorMsgs");
          for (String message : errorMsgs) {
            <li><%=message%></li>
          }
        %>
      </ul>
      <a href='AddLeagueForm'> Back up and try again.</a>
    </font>
    </p>
  </body>
</html>

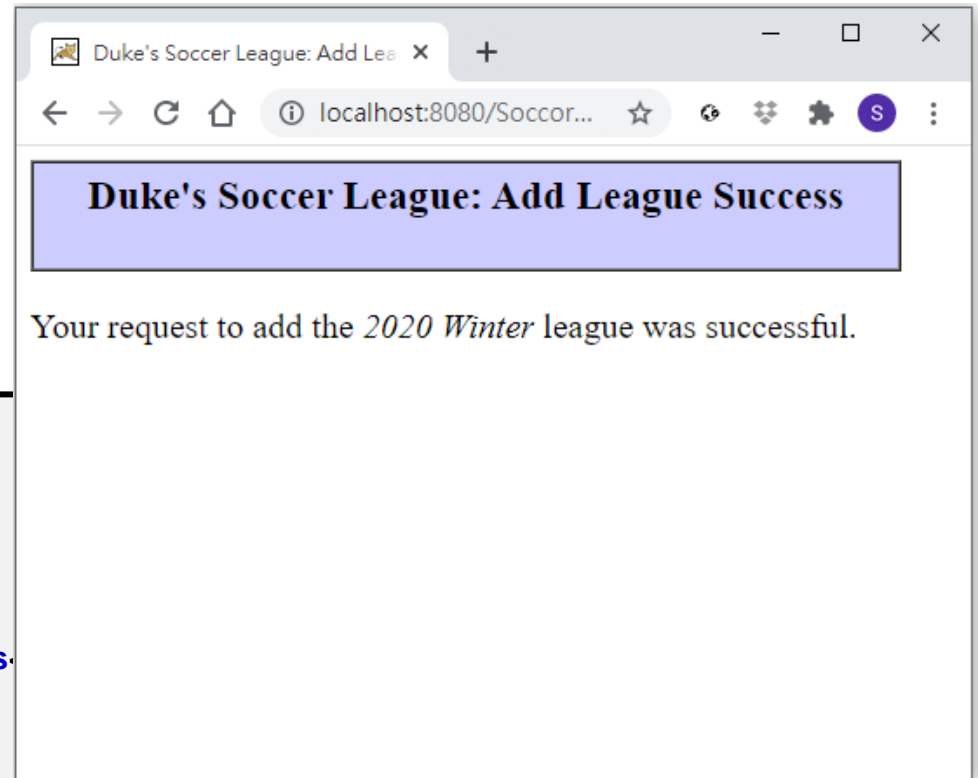
```



Soccer 範例

```
<%@ page contentType="text/html"%>
<%@ page session="false" %>
<%@ page import="model.League" %>

<html>
  <head>
    <title>Duke's Soccer League: Add League Success</title>
  </head>
  <body bgcolor='white'>
    <!-- Page Heading -->
    <table border='1' cellpadding='5' cellspacing='0' width='400'>
      <tr bgcolor='#CCCCFF' align='center' valign='center' height='20'>
        <td><h3>Duke's Soccer League: Add League Success</h3></td>
      </tr>
    </table>
    <%-- Generate main body --%>
    <p>
      Your request to add the <i>${league.title}</i> league was successful.
    </p>
  </body>
</html>
```



Lab 2 – 表單驗證



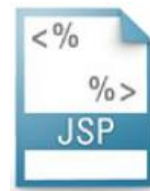
Library User

新增 DVD 加入清單



Library User

<http://localhost:8080/DVDLibrary/AddDVDForm>



addDVD.jsp

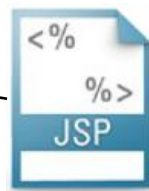
http://localhost:8080/DVDLibrary/Add_DVD



AddDVDServlet



DVD



error.jsp



success.jsp

Lab 2

- 撰寫 `controller.AddDVDServlet.java`
 - Servlet URL 為 `Add_DVD`
 - 取得表單中請求參數 `title`, `year`, `genre`, `otherGenre`
 - 若使用者輸入 `otherGenre`, `genre=otherGenre`
 - 若使用者未輸入 `otherGenre`, `genre` 使用下拉選單值
 - 表單驗證
 - 宣告一個集合來儲存錯誤訊息
 - `title` 不可為空值(`null`或長度為0)
 - 否則在錯誤訊息集合中加入"請輸入DVD標題" 字串
 - `year` 不可為空值(`null`或長度為0)
 - 否則在錯誤訊息集合中加入"請輸入DVD發行年度" 字串
 - `year` 不可為無效數字值 (使用字串的`matches("\\d\\d\\d\\d")`方法)
 - 否則在錯誤訊息集合中加入"DVD發行年度請輸入有效數字" 字串

Lab 2

- `controller.AddDVDServlet.java`

- 表單驗證失敗

- 錯誤訊息集合不是空的
 - 將錯誤訊息集合加入請求屬性 `errorMsgs`
 - 將請求轉送至`error.jsp`

- 表單驗證成功

- 以`title, year, genre`建立`DVDItem`物件
 - 將`DVDItem`物件加入請求屬性`dvdItem`
 - 將請求轉送至`success.jsp`

Lab 2

■ 撰寫error.jsp

- ❑ 使用JSP scripting 元素
- ❑ 取得errorMsgs屬性
- ❑ 將錯誤訊息顯示於網頁如右圖

■ 撰寫success.jsp

- ❑ 使用JSP scripting 元素
- ❑ 取得dvdItem屬性
- ❑ 將DVD資訊顯示於網頁如右圖

■ 測試、執行

