

---

# Java EE7 Web Session 管理

---

鄭安翔

ansel\_cheng@hotmail.com

---

# 課程大綱

- 1) **HTTP 與 Session 管理**
- 2) Session 管理實作

# Soccer 範例



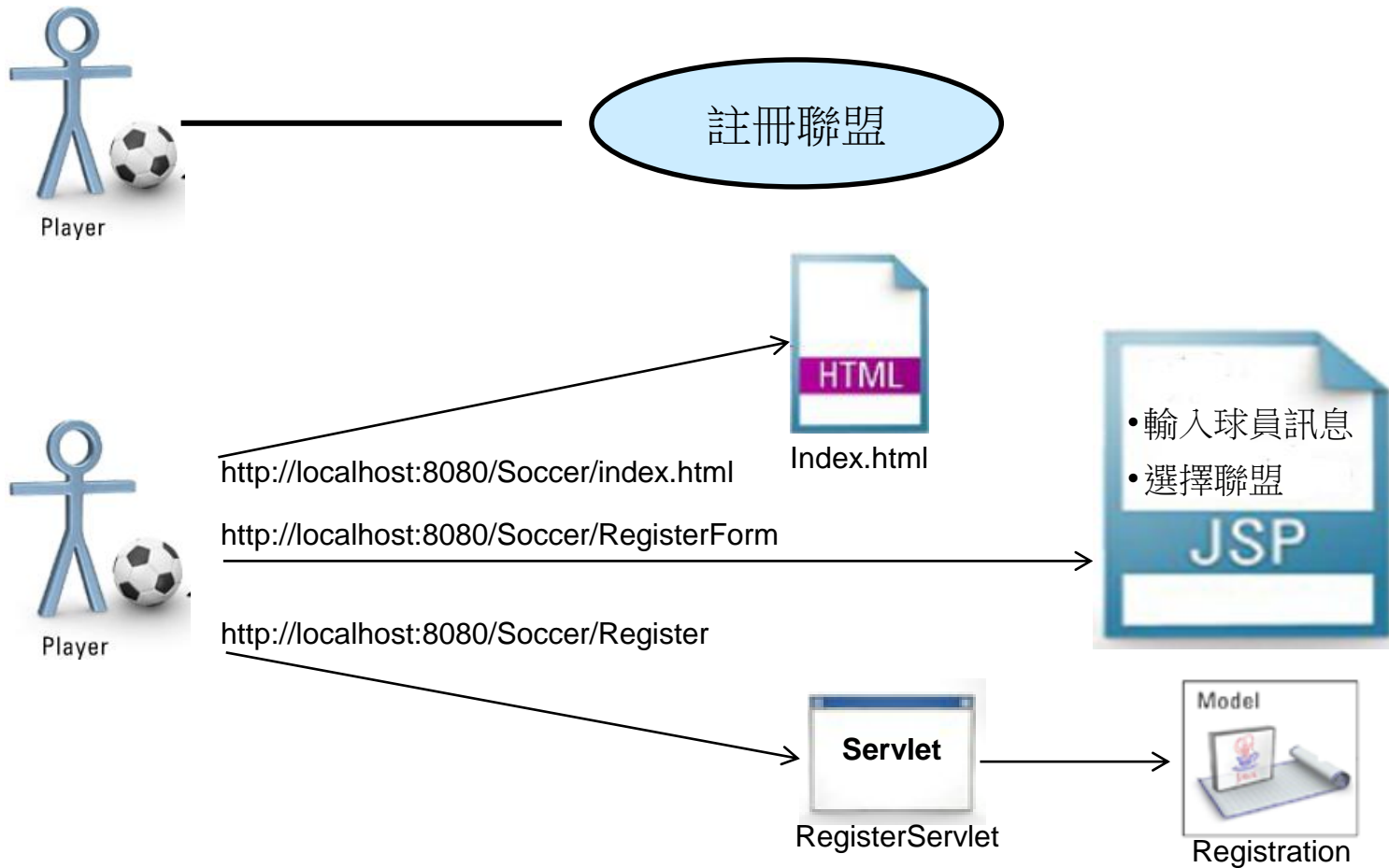
## 足球聯盟網路應用程式

列出所有可用的聯盟

註冊聯盟

增加新聯盟

# Soccer 範例



localhost:8080/Soccer/registra x +

localhost:8080/Socc... ☆ ⚙ S

## Duke's Soccer League: Registration

This form allow you to register a soccer league.

**Enter User Information :**

Name:

Address:

City:

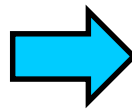
Province:

Postal code:

**Enter User Information :**

Year:

Season:



Duke's Soccer League: Registre x +

localhost:8080/Socc... ☆ ⚙ S

## Duke's Soccer League: Registration

1) Enter Player Info	2) Select League
----------------------	------------------

Name:

Address:

City:

Province:

Postal code:

Duke's Soccer League: Registre x +

localhost:8080/Socc... ☆ ⚙ S

## Duke's Soccer League: Registration

1) Enter Player Info	2) Select League
----------------------	------------------

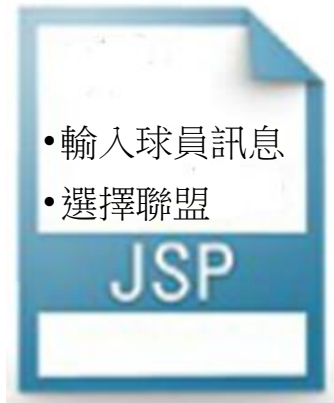
Year:

Season:

# Soccer 範例



Player

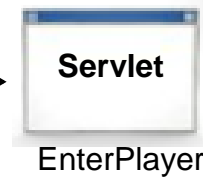


- 輸入球員訊息
- 選擇聯盟

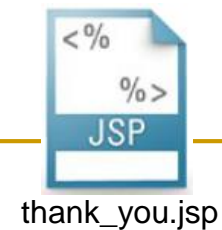
[http://localhost:8080/Soccer/enter\\_player.jsp](http://localhost:8080/Soccer/enter_player.jsp)



<http://localhost:8080/Soccer/EnterPlayer>

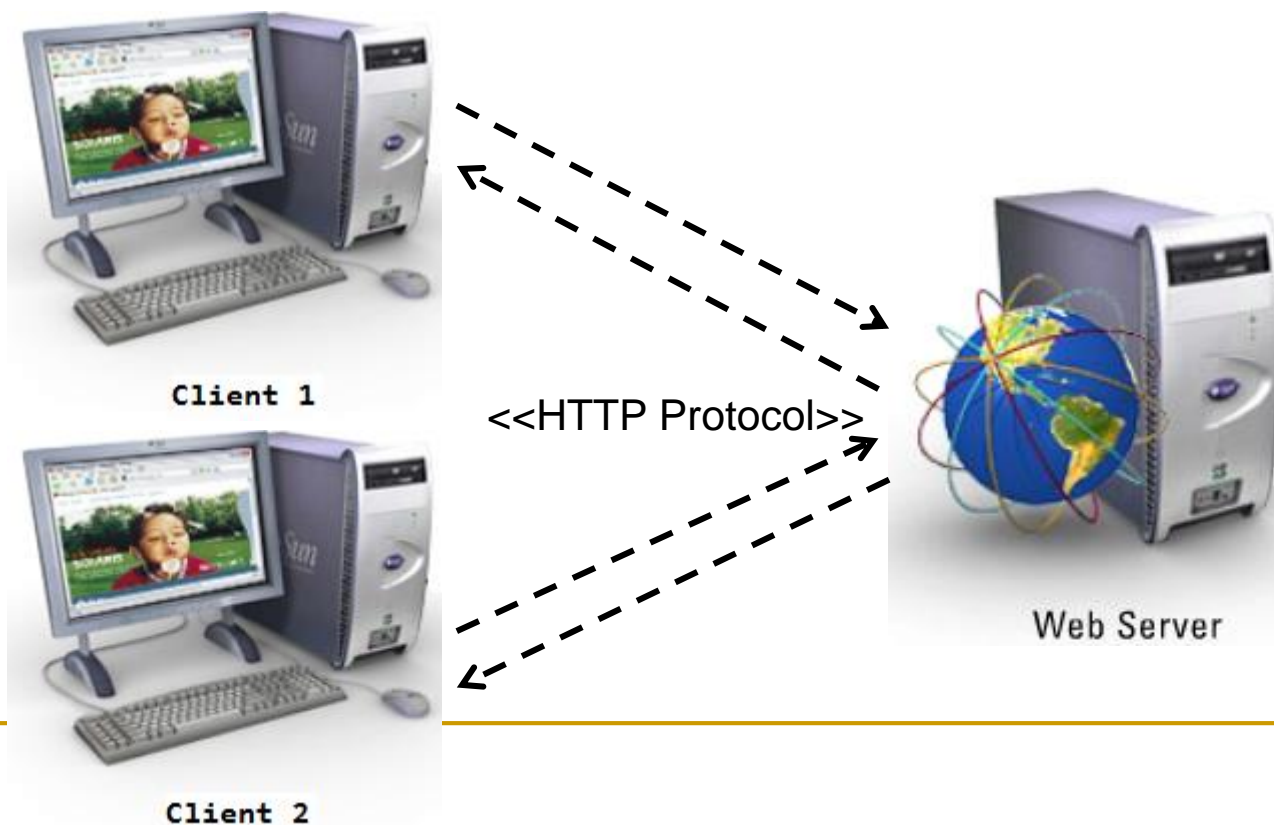


<http://localhost:8080/Soccer/SelectLeague>



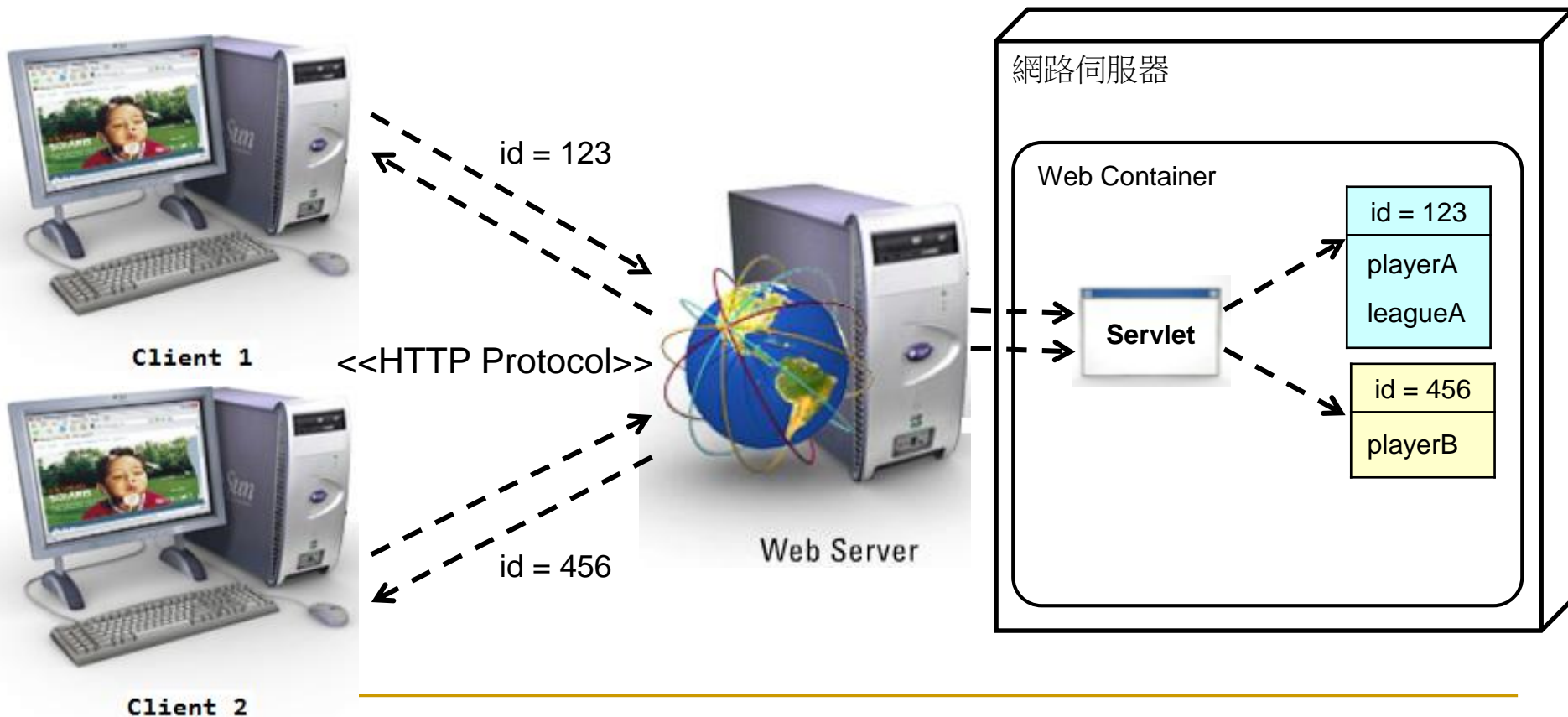
# HTTP是無狀態通訊協定

- 每個請求與回應訊息的連線都是獨立的。
- 使用者連續發出兩個HTTP請求,伺服器不會紀錄前一個請求的操作結果



# Session 機制

- 網路元件容器提供Session機制
  - 解決HTTP Protocol 無狀態的問題





# javax.servlet.http.HttpSession

## <<interface>> HttpServletRequest

getContextPath() : String  
getCookies() : Cookie[]  
getDateHeader(name : String) : long  
getHeader(name : String) : String  
getHeaderNames() : Enumeration  
getHeaders() : Enumeration  
getIntHeader(name : String) : int  
getMethod() : String  
getQueryString() : String  
getRemoteUser() : String  
getRequestedSessionId() : String  
getRequestURI() : String  
getRequestURL() : String  
getSession() : HttpSession  
getSession(create : boolean) : HttpSession  
isUserInRole(role : String) : boolean

## <<interface>> HttpSession

setAttribute(name : String, value : Object)  
getAttribute(name : String) : Object  
getAttributeNames() : Enumeration  
removeAttribute(name : String)  
invalidate()  
getMaxInactiveInterval() : int  
setMaxInactiveInterval(interval : int)  
getCreationTime() : long  
getLastAccessTime() : long  
getServletContext() : ServletContext  
isNew() : boolean  
getID() : String

# getSession method

## ■ HttpServletRequest

### □ getSession()

- 傳回Request中的Session物件
- 若Request中不存在Session 物件,Container建構一個新的 Session 物件傳回.

### □ getSession(boolean create)

- true : 同getSession()
- false :若session 物件不存在,傳回Null

# HttpSession

- Session建立Web Application,與Web Browser 之間關係
  - 一個Context,對同一個使用者,只會建立一個Session物件
  - Context中不同的use case會共用session物件
  - 使用者同時用兩種瀏覽器瀏覽同一個網站,會有兩個不同的Session

# Session 範圍的共用資料

- 同一Session(使用者) 共用資料存取
  - `setAttribute(name : String, value : Object)`
  - `getAttribute(name : String) : Object`
  - `removeAttribute(name : String)`
    - 命名時盡量清楚,小心不同物件名稱重複
    - 不再使用的資源應作 **remove**
    - `setAttribute`時,屬性已存在,將更新屬性值
    - `getAttribute`時若無對應屬性,傳回null

# JSP 中使用Session共享資源

## ■ JSP指令標籤設定Session使用

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

## ■ JSP Expression / Scriptlet Tag 中

### □ 使用隱含變數Implicit Variables

變數名稱	型別	用途	Scope
<b>session</b>	javax.servlet.http. HttpSession	共享相同 HTTP 連線階段 (相同使用者)內容	Session

```
<%= session.getAttribute("attributeName" ) %>
```

```
<%
```

```
    Type obj = (Type) session.getAttribute("attributeName");
```

```
    .....
```

```
%>
```

# JSP 中使用Session共享資源

- 使用表示式語言 (EL)
  - 未指定變數範圍時,自動尋找變數
  - 自動轉變資料型態

`$\${attributeName}$`

`$\${sessionScope.attributeName}$`

# Destroying the Session

## ■ Session物件回收

- 程式中使用 *invalidate()* 回收 Session
  - 由Web container管理閒置時間
    - Container 追蹤 Session 的 inactive interval 閒置時間
      - 超過設定之最大閒置時間, 容器回收(invalidate) session
    - 設定 Context 中所有 sessions 的 timeout 長度
      - web.xml 中 session-timeout 標籤
    - 程式中修改 session 的 timeout 長度
- HttpSession
- *setMaxInactiveInterval(seconds : int)*
  - *getMaxInactiveInterval() : int*

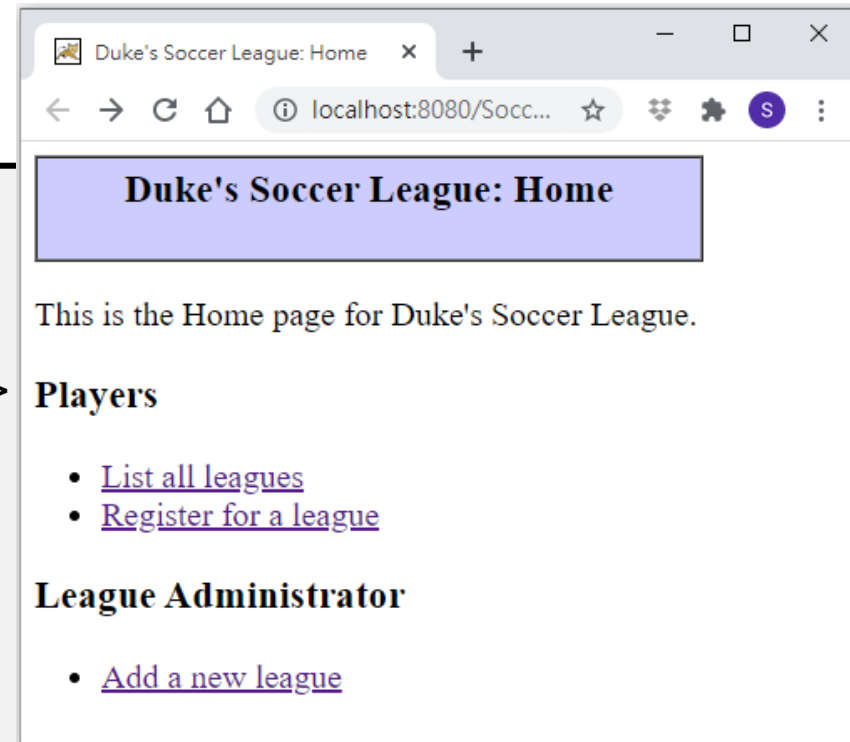
# Destroying the Session

```
<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>
  </servlet>
  <servlet-mapping>
    <servlet-name> Servlet 名稱. </servlet-name>
    <url-pattern> Servlet URL 對應. </url-pattern>
  </servlet-mapping>
  <session-config>
    <session-timeout>Timeout時間(單位:分鐘)</session-timeout>
  </session-config>
</web-app>
```



# Soccer 範例 – index.html

```
<html>
<head>
  <title>Duke's Soccer League: Home</title>
</head>
<body>
  <table border='1' cellpadding='5' cellspacing='0' width='400'>
    <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='list_leagues.jsp'>List all leagues</a></li>
    <li><a href='enter_player.jsp'>Register for a league</a></li>
  </ul>
  <h3>League Administrator</h3>
  <ul>
    <li><a href='admin/AddLeagueForm'>Add a new league</a></li>
  </ul>
</body>
</html>
```



# Soccer 範例-Domain/Service Objects

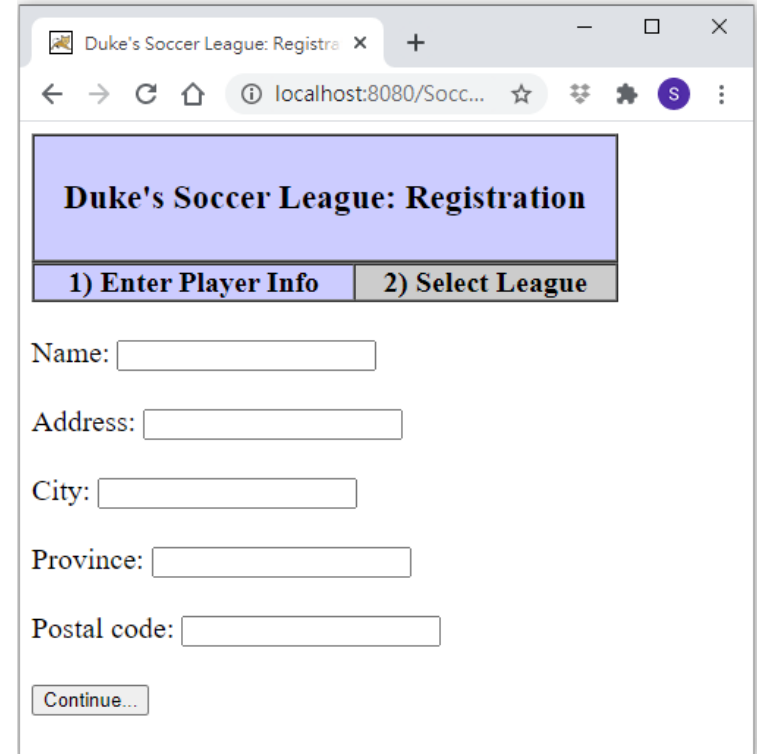
```
package model;
public class Player {
    String name;
    String address;
    String city;
    String province;
    String postalCode;
    public Player(String name, String address, String city,
        String province, String postalCode) {
        this.name = name;
        this.address = address;
        this.city = city;
        this.province = address;
        this.postalCode = postalCode;
    }
    public int getName() {
        return name;
    }
    public String getAddress() {
        return address;
    }
    public String getCity() {
        return city;
    }
    public String getProvince() {
        return province;
    }
    public String getPostalCode() {
        return postalCode;
    }
}
```

```
package model;
public class RegistrationService {
    private League league;
    private Player player;
    public RegistrationService(League league, Player player) {
        this.league = league;
        this.player = player;
    }
    public void setLeague(League league) {
        this.league = league;
    }
    public void setPlayer(Player player) {
        this.player = player;
    }
    public League getLeague() {
        return league;
    }
    public Player getPlayer() {
        return player;
    }
    public void register(){
        System.out.println("creating Player : "+player.getName() );
        System.out.println("Register to " + league.getTitle()
            + " League!");
    }
}
```

# Soccer 範例 - enter\_player.jsp

```
<!DOCTYPE html>
<%@ page contentType="text/html"%>

<html>
  <head>
    <title>Duke's Soccer League: Registration</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: Registraion</h3></td>
      </tr>
    </table>
    <table border='1' cellpadding='0' cellspacing='0' width='400'>
      <tr align='center' valign='center' height='20'>
        <td bgcolor='#CCCCFF' ><b>1) Enter Player Info</b></td>
        <td bgcolor='#CCCCCC' ><b>2) Select League</b></td>
      </tr>
    </table><br/>
    <form action='EnterPlayer' method='POST'>
      Name: <input type='text' name='name' /> <br/><br/>
      Address: <input type='text' name='address' /> <br/><br/>
      City: <input type='text' name='city' /> <br/><br/>
      Province: <input type='text' name='province' /> <br/><br/>
      Postal code: <input type='text' name='postalCode' /> <br/><br/>
      <input type='Submit' value='Continue...' />
    </form>
  </body>
</html>
```



The screenshot shows a web browser window with the title "Duke's Soccer League: Registra...". The address bar shows "localhost:8080/Socc...". The page content includes a header "Duke's Soccer League: Registration" and two tabs: "1) Enter Player Info" (selected) and "2) Select League". Below the tabs, there are input fields for "Name:", "Address:", "City:", "Province:", and "Postal code:". At the bottom, there is a "Continue..." button.

# Soccer 範例 – EnterPlayer.Java

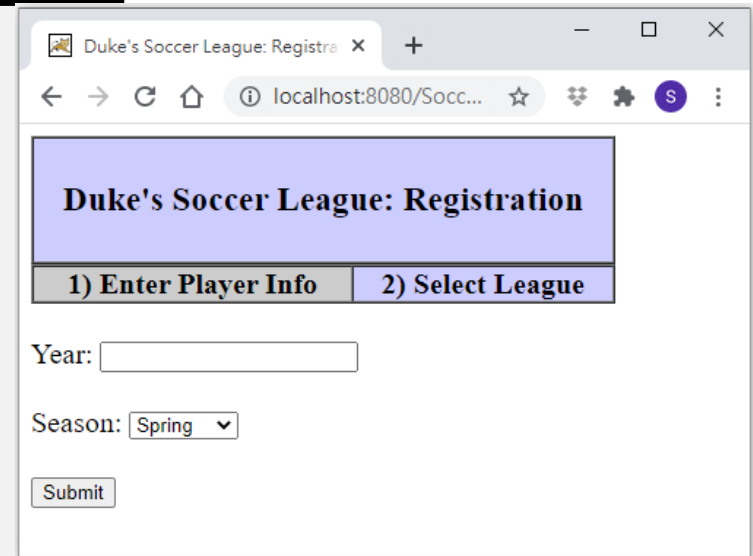
```
public void processRequest (HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    // Keep a set of strings to record form processing errors.
    List<String>errorMsgs = new LinkedList<String> ();
    // Store this set in the request scope, in case we need to
    // send the ErrorPage view.
    request.setAttribute("errorMsgs", errorMsgs);
    ServletContext context = this.getServletContext();
    try {
        // Retrieve form parameters.
        String name = request.getParameter("name").trim();
        String address = request.getParameter("address").trim();
        String city = request.getParameter("city").trim();
        String province = request.getParameter("province").trim();
        String postalCode = request.getParameter("postalCode").trim();
        // Verify 'Enter Player Information' form fields
        if ( name.length() == 0 ) {
            errorMsgs.add("You must enter your full name.");
        }
        if ( (address.length() == 0) || (city.length() == 0)
            || (province.length() == 0) || (postalCode.length() == 0) ) {
            errorMsgs.add("You must enter your full address.");
        }
    }
```

```
        // Send the ErrorPage if there were errors
        if ( ! errorMsgs.isEmpty() ) {
            RequestDispatcher view =
                request.getRequestDispatcher("error.jsp");
            view.forward(request, response);
            return;
        }
        // Perform business logic
        // Store the Player object in the session-scope
        Player player = new Player(name, address, city,
                                    province, postalCode);
        HttpSession session = request.getSession();
        session.setAttribute("player", player);
        // Send the Success page
        RequestDispatcher view =
            request.getRequestDispatcher("select_league.jsp");
        view.forward(request, response);

    } catch (RuntimeException e) {
        errorMsgs.add(e.getMessage());
        RequestDispatcher view =
            request.getRequestDispatcher("error.jsp");
        view.forward(request, response);
        // Log stack trace
        this.log(e.toString());
    }
}
```

# Soccer 範例 - select\_league.jsp

```
<!DOCTYPE html>
<%@ page contentType="text/html"%>
<%@ page import="java.util.*, model.*"%>
<html>
  <head>
    <title>Duke's Soccer League: Registration</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: Registration</h3></td>
      </tr>
    </table>
    <table border='1' cellpadding='0' cellspacing='0' width='400'>
      <tr align='center' valign='center' height='20'>
        <td bgcolor='#CCCCCC' ><b>1) Enter Player Info</b></td>
        <td bgcolor='#CCCCFF' ><b>2) Select League</b></td>
      </tr>
    </table><br/>
    <form action='SelectLeague' method='POST'>
      Year: <input type='text' name='year'/> <br/><br/>
      Season: <select name='season'/>
      <% String[] SEASONS =(String[])application.getAttribute("seasonList");
      for ( int i = 0; i < SEASONS.length; i++ ) {    %>
        <option value='<%= SEASONS[i] %>'> <%= SEASONS[i] %> </option>
      <% } // END of for loop over seasons array %>
      <input type='Submit' value='Submit' />
    </form>
  </body>
</html>
```



```

public 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 it
    // send the ErrorPage view.
    request.setAttribute("errorMsgs", errorMsgs);
    ServletContext context = this.getServletContext();
    try {
        // Retrieve form parameters.
        String yearStr = request.getParameter("year").trim();
        String season = request.getParameter("season").trim();
        // Perform data conversions.
        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 be within 2010-2025");
        }
        if ( season.equals("UNKNOWN") ) {
            errorMsgs.add("Please select a league season");
        }
        League league = this.findLeague(year, season);
        if ( league==null ) {
            errorMsgs.add("Please enter valid league.");
        }
    }
}

```

```

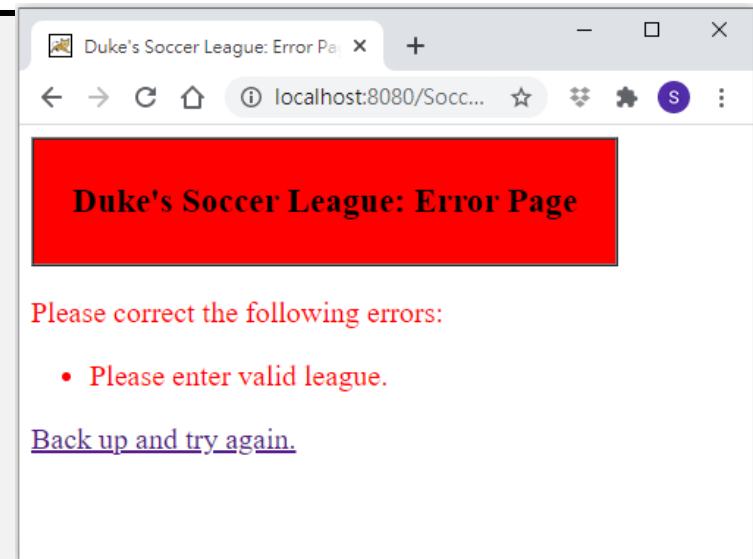
    // Send the ErrorPage if there were errors
    if ( ! errorMsgs.isEmpty() ) {
        RequestDispatcher view = request.getRequestDispatcher("error.jsp");
        view.forward(request, response);
        return;
    }
    // Perform business logic
    HttpSession session = request.getSession();
    session.setAttribute("league", league);
    Player player = (Player)session.getAttribute("player");
    RegistrationService registration = new RegistrationService(league, player);
    registration.register();
    // Send the Success page
    RequestDispatcher view = request.getRequestDispatcher("thank_you.jsp");
    view.forward(request, response);
} catch (RuntimeException e) {
    errorMsgs.add(e.getMessage());
    RequestDispatcher view = request.getRequestDispatcher("error.jsp");
    view.forward(request, response);
}
}

private League findLeague(int year, String season){
    ServletContext context = getServletContext();
    List<League> leagueList = (List<League>)
        context.getAttribute("leagueList");
    for(League league : leagueList){
        if(league.getYear()==year && league.getSeason().equals(season))
            return league;
    }
    return null;
}
}

```

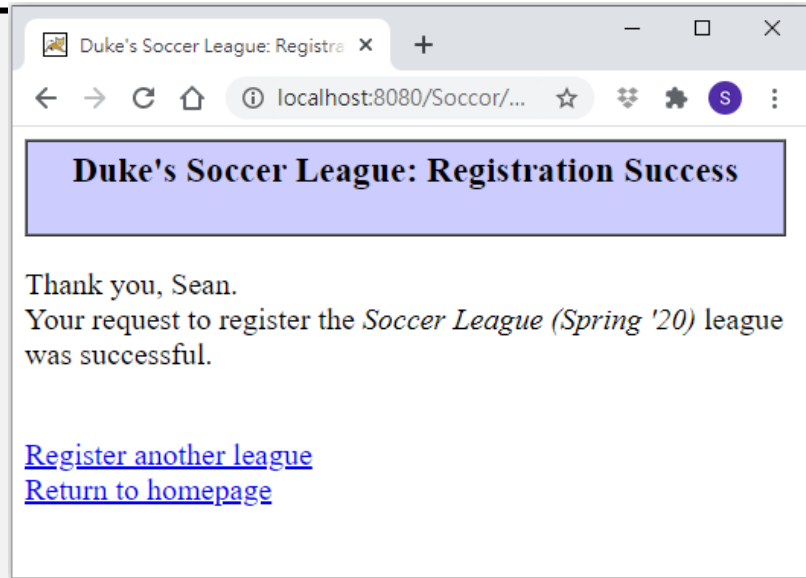
# Soccer 範例 - error.jsp

```
<!DOCTYPE html>
<%@ page contentType="text/html"%>
<%@ page import="java.util.*"%>
<%@ page isELIgnore = "false"%>
<html>
  <head>
    <title>Duke's Soccer League: Error Page</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: Error Page</h3></td>
      </tr>
    </table>
    <p><font color='red'>Please correct the following errors:</font><ul>
    <% List<String> errorMsgs = (List<String>) request.getAttribute("errorMsgs");
      for (String message : errorMsgs) %>
        <li><%=message%></li>
    <% } %>
    </ul>
    <a href='enter_player.jsp'>Back up and try again.</a>
    </font> </p>
  </body>
</html>
```



# Soccer 範例 – thank\_you.jsp

```
<!DOCTYPE html>
<%@ page contentType="text/html"%>
<%@ page session="true"%>
<%@ page import="model.*"%>
<%
    League league = (League) session.getAttribute("league");
    Player player = (Player) session.getAttribute("player");
%>
<html>
<head>
    <title>Duke's Soccer League: Registration Success</title>
</head>
<body bgcolor='white'>
<!-- Page Heading -->
<table border='1' cellpadding='5' cellspacing='0' width='500'>
    <tr bgcolor='#CCCCFF' align='center' valign='center' height='20'>
        <td><h3>Duke's Soccer League: Registration Success </h3></td>
    </tr>
</table>
<p>
    Thank you, <%=player.getName()%> <br/>
    Your request to register the <i><%=league.getTitle()%></i> league was successful.
</p><br/>
<a href='select_league.jsp'> Register another league.</a>
<a href='index.html'> Return to homepage.</a>
</body>
</html>
```





# Soccer 範例-InitializeLeagues.java

```
package web;
import javax.servlet.*, java.util.*, java.io.*, javax.servlet.annotation.*, model.League;
@WebListener
public class InitializeLeagues implements ServletContextListener {
    @Override
    public void contextInitialized(ServletContextEvent sce) {
        ServletContext context = sce.getServletContext();
        String[] SEASONS = {"Spring", "Summer", "Autumn", "Winter"};
        String seasonStr = context.getInitParameter("season-list");
        if((seasonStr != null) || (seasonStr.length() != 0))
            SEASONS = seasonStr.split(", ");
        context.setAttribute("seasonList", SEASONS);
        context.log("The season list has been loaded.");
        .....
    }
    @Override
    public void contextDestroyed(ServletContextEvent sce) {
    }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app>
    <context-param>
        <param-name>season-list</param-name>
        <param-value>Spring, Summer, Autumn, Winter</param-value>
    </context-param>
    <context-param>
        <param-name>leaguesFile</param-name>
        <param-value>/WEB-INF/data/leagues.txt</param-value>
    </context-param>
    <listener>
        <description>ServletContextListener</description>
        <listener-class>web.InitializeLeagues</listener-class>
    </listener>
</web-app>
```

# 課程大綱

- 1) HTTP 與 Session 管理
- 2) **Session 管理實作**
  - **Cookie**
  - **URL Rewriting**

# Session Management

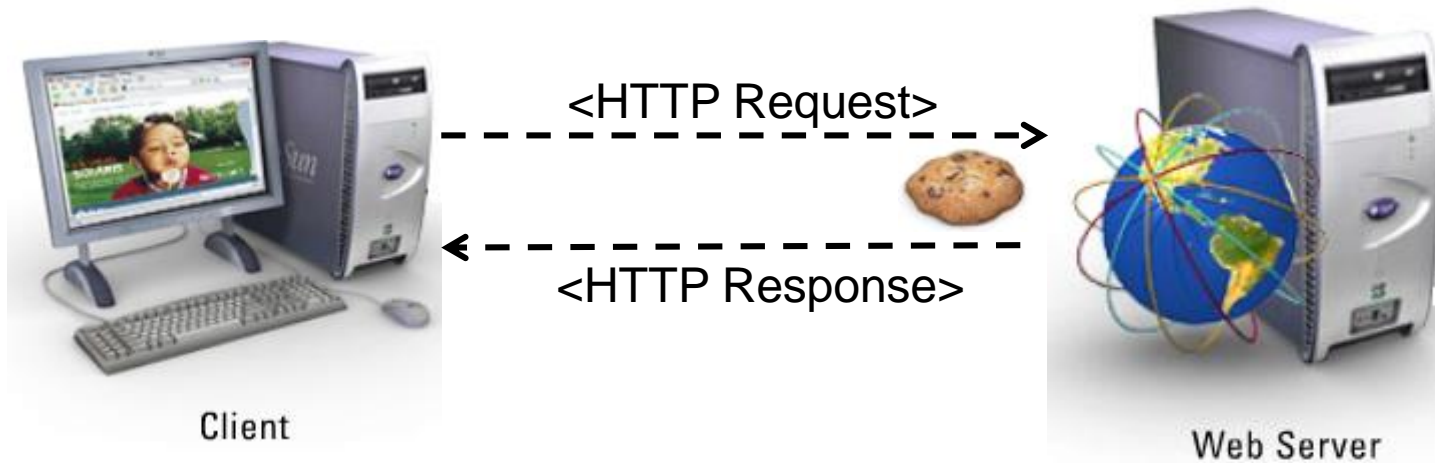
## ■ Session 管理實作

- 網路元件容器(container)廠商必需支援session機制
- 元件容器給每個客戶(Client)一個不同的SeesionID
- 元件容器可依SessionID,建立代表該客戶的Session物件
- 程式中要取得Session物件時,元件容器依請求(Request)中的session id,找到對應的Session物件
- 取得 SessionID
  - Cookie
  - URL Rewriting
  - 使用SSL encryption key作為Session ID

# Cookie

## ■ Cookies

- 讓網路伺服器可以在用戶端機器儲存資訊。
- HTTP 的擴充版本
  - Internet Engineering Task Force (IETF) Request for Comment (RFC) 2109 定義

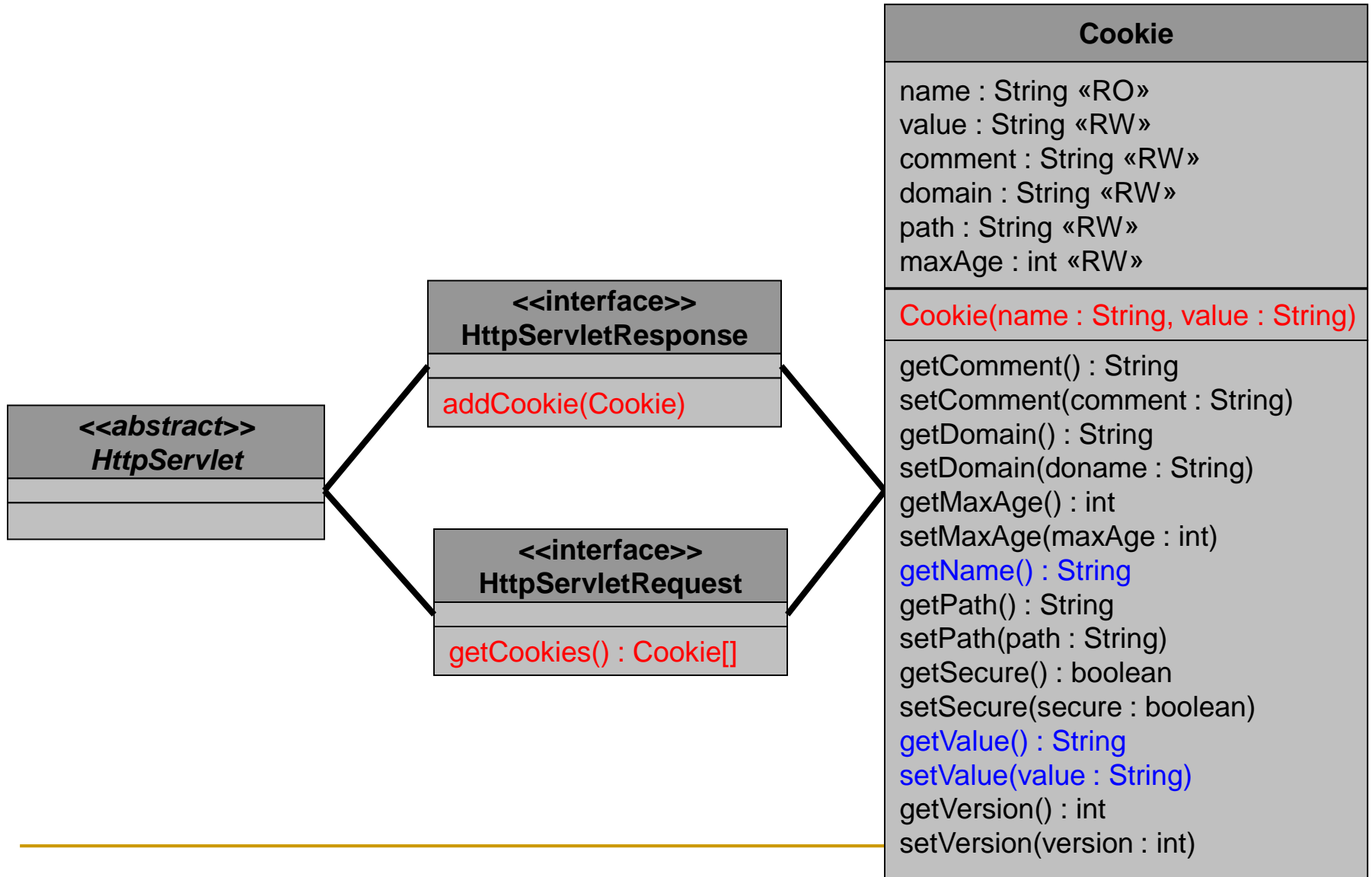


# Cookie

## ■ Cookies 特性

- 儲存在瀏覽器用戶端電腦中的鍵值對資訊
  - 由網路伺服器在回應**Response**中建立及更新
  - 瀏覽器依 **Response** 訊息,將資料存入用戶端電腦系統中
- 儲存時依網路伺服器的網域名稱區隔,儲存在不同檔案
  - 可再進一步用該網域下的路徑來區隔。
- 瀏覽器發送請求時,將該網域名稱相關的所有**cookies**發送到伺服器
- **cookie** 有使用期限, 在使用期限結束時會被用戶端的瀏覽器清除。

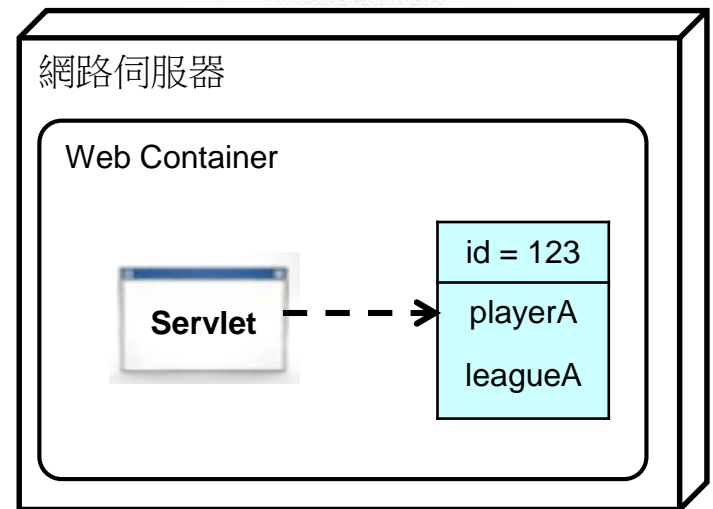
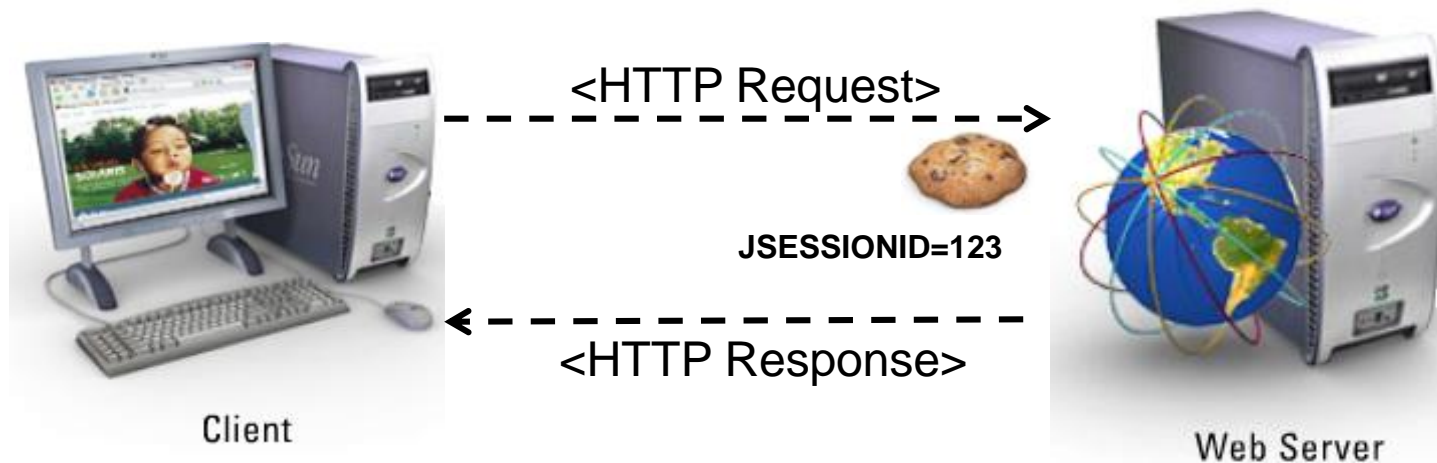
# Cookie



# Session ID 傳遞 - Cookie

- 使用Cookie傳遞Session ID
  - 在Client電腦中存入JSESSIONID的Cookies
  - 程式不需下任何指令
    - 只要程式使用Session物件,元件容器自動在回應(Response)中加入JSESSIONID的Cookies
    - 以後每一個Request皆依此辨別身分
  - 缺點
    - 瀏覽器不一定支援
    - 瀏覽器可以關閉Cookie功能

# Session ID 傳遞 - Cookie





# Session ID 傳遞 – URL Rewriting

## ■ 使用URL-Rewriting傳遞Session ID

- Request路徑後加上session id

  - Action = 'URL' → 'URL ; jsessionid=xxx'

- jsessionid由元件容器動態產生

HttpResponse 物件的兩個方法

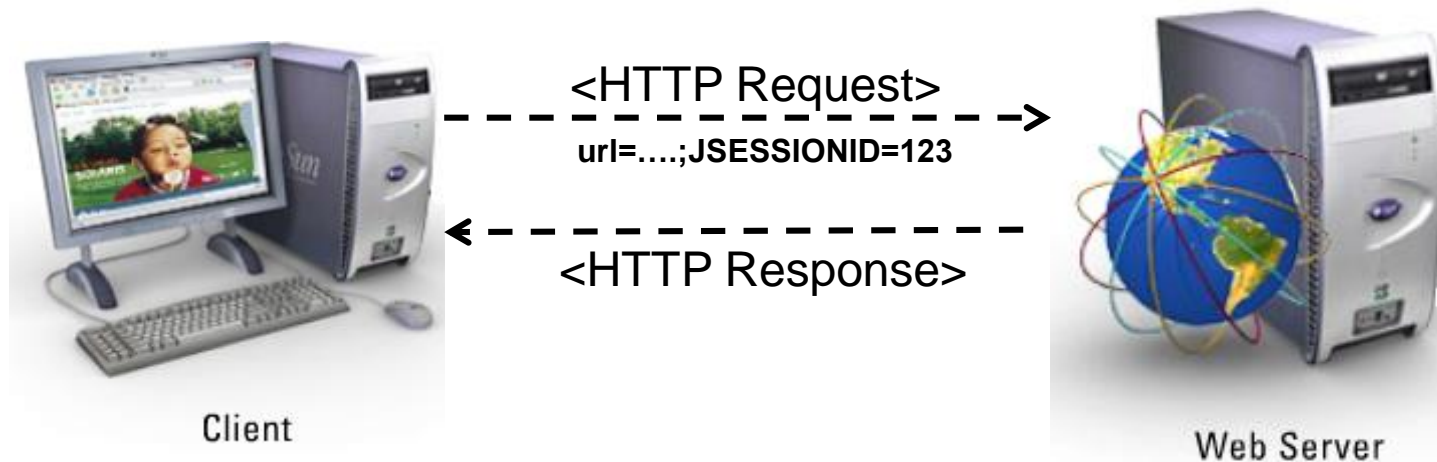
encodeURL(url : String) : String

encodeRedirectURL(url : String) : String

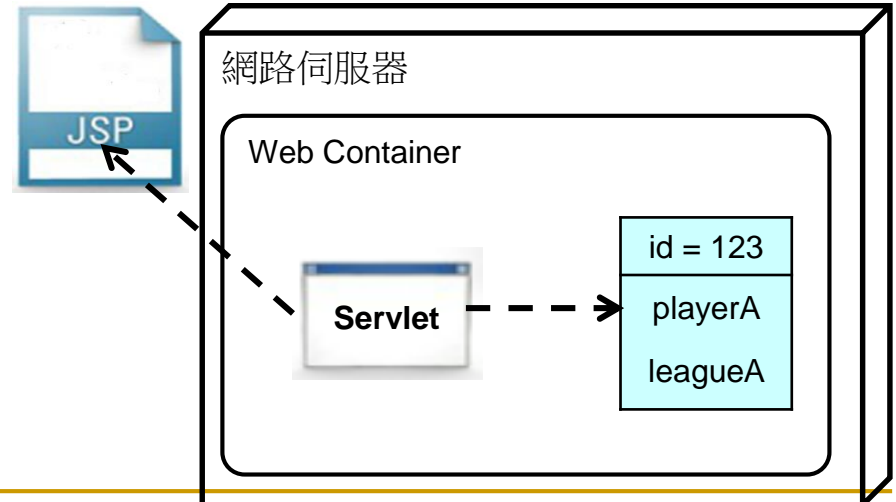
- View 頁面需用Servlet或JSP產生

- 當cookie功能開啟時,使用cookie傳遞Session ID

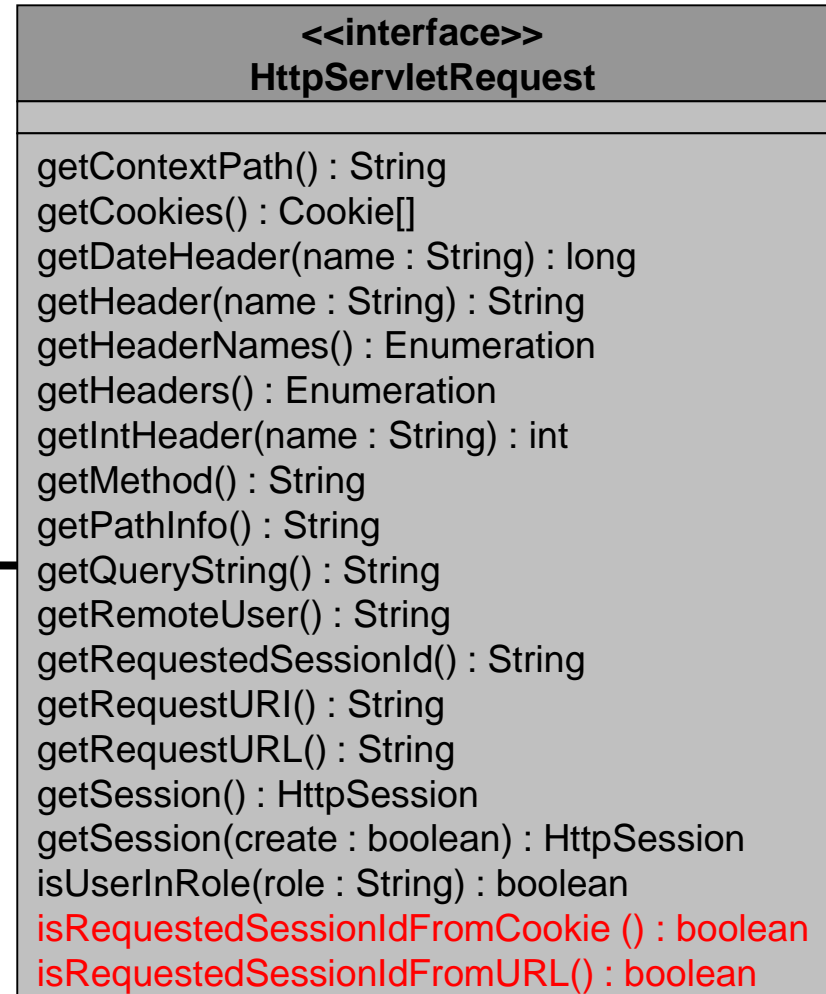
# Session ID 傳遞 - URL Rewriting



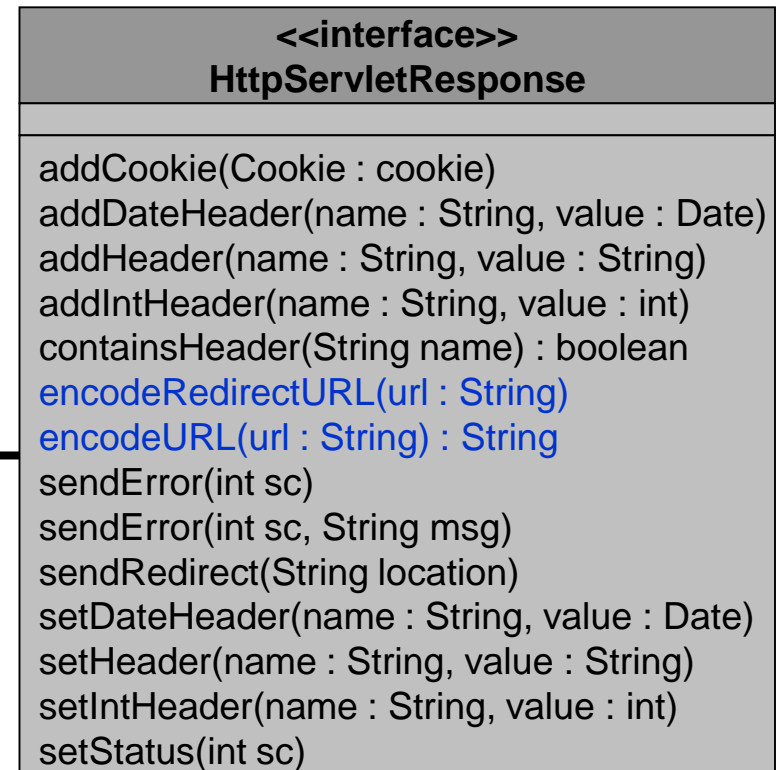
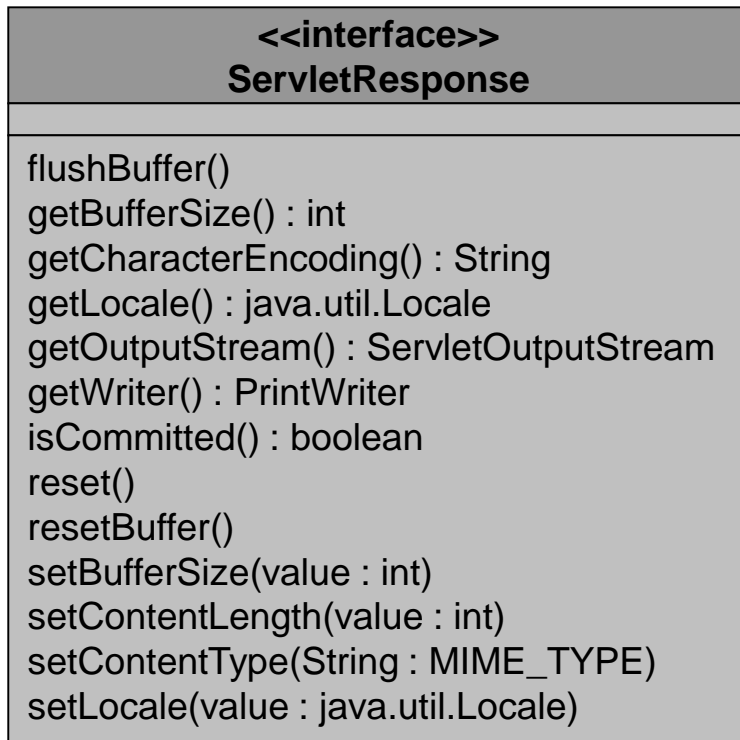
`url=...;JSESSIONID=123`



# HttpServletRequest – Session ID



# HttpServletResponse - URL Rewriting



# Soccer 範例

enter\_player.jsp

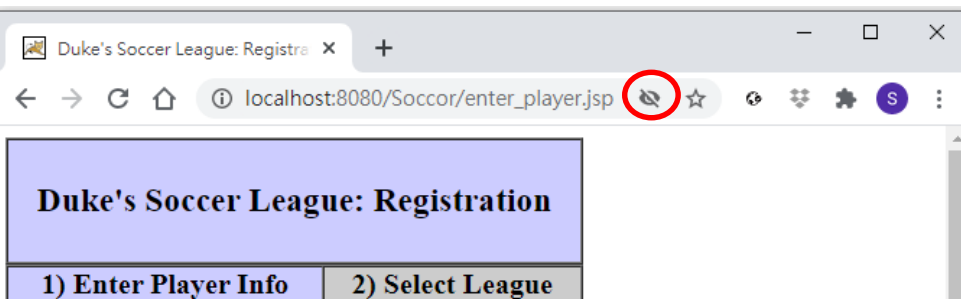
```
.....
<form action='<%=response.encodeURL("EnterPlayer")%>' method='POST'>
  Name: <input type='text' name='name' /> <br/><br/>
  Address: <input type='text' name='address' /> <br/><br/>
  City: <input type='text' name='city' /> <br/><br/>
  Province: <input type='text' name='province' /> <br/><br/>
  Postal code: <input type='text' name='postalCode' /> <br/><br/>
  <input type='Submit' value='Continue...' />
</form>
</body>
</html>
```

select\_league.jsp

```
.....
<form action='<%=response.encodeURL("SelectLeague")%>' method='POST'>
  Year: <input type='text' name='year' /> <br/><br/>
  Season: <select name='season' />
  <% String[] SEASONS =(String[])application.getAttribute("seasonList");
    for ( int i = 0; i < SEASONS.length; i++ ) {    %>
      <option value='<%= SEASONS[i] %>'> <%= SEASONS[i] %> </option>
  <% } // END of for loop over seasons array %>
  <input type='Submit' value='Submit' />
</form>
</body>
</html>
```

thank\_you.jsp

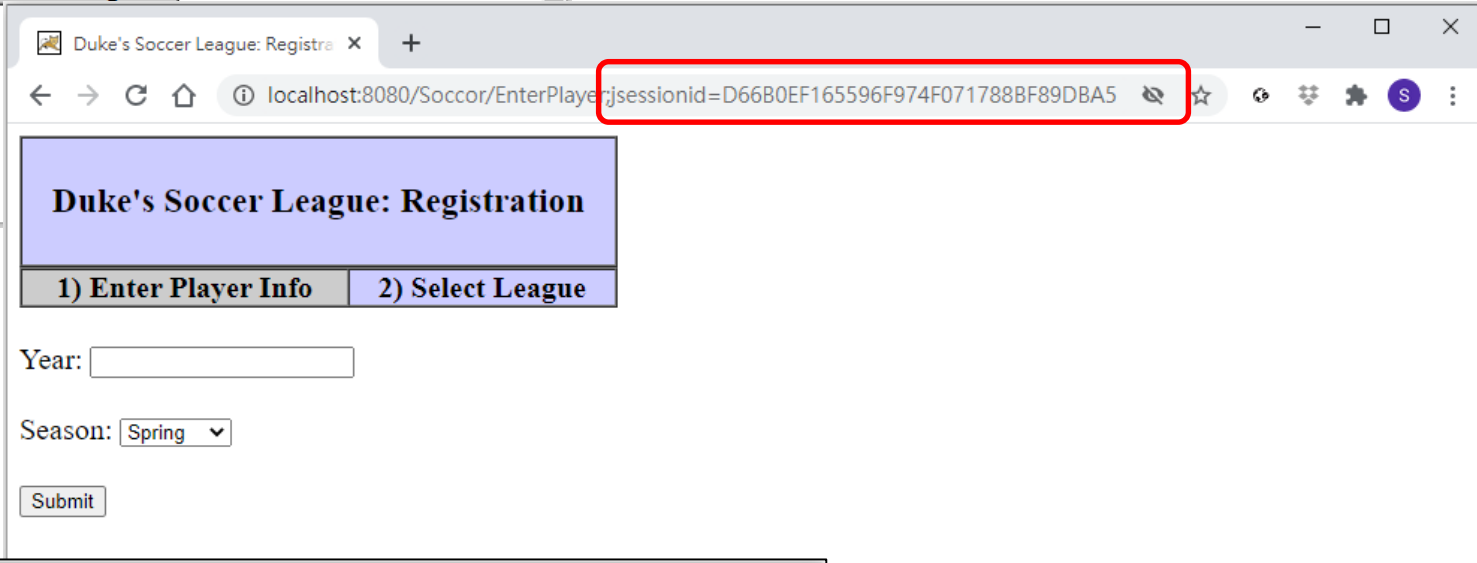
```
.....
<p>
  Your request to register the <i> </i> league was successful.
</p><br/>
<a href='<%=response.encodeURL("select_league.jsp")%>'> Register another league.</a>
<a href='index.html'> Return to homepage.</a>
</body>
</html>
```



Name:

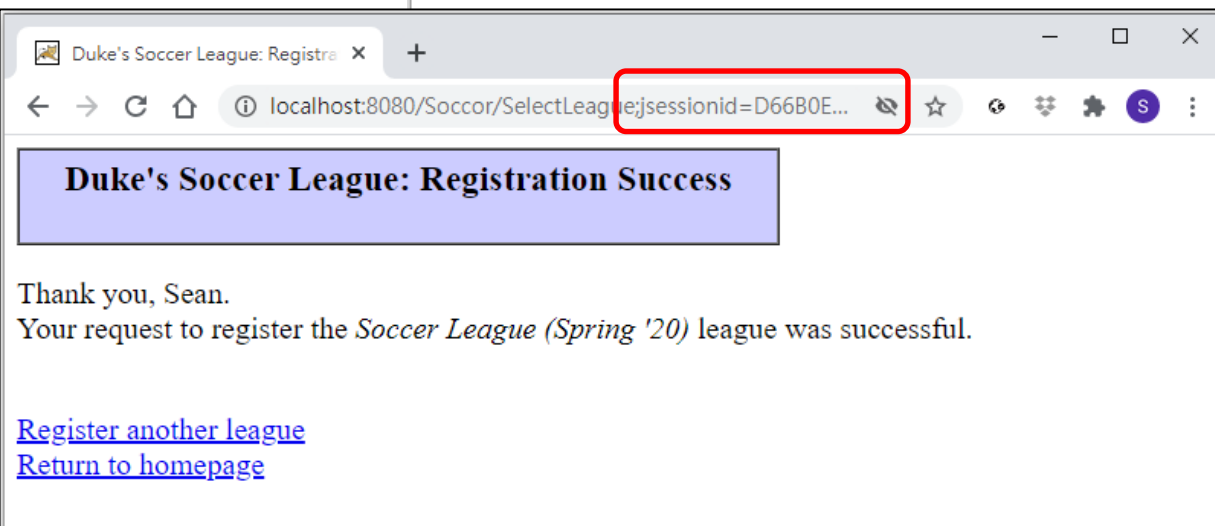
Address:

City:

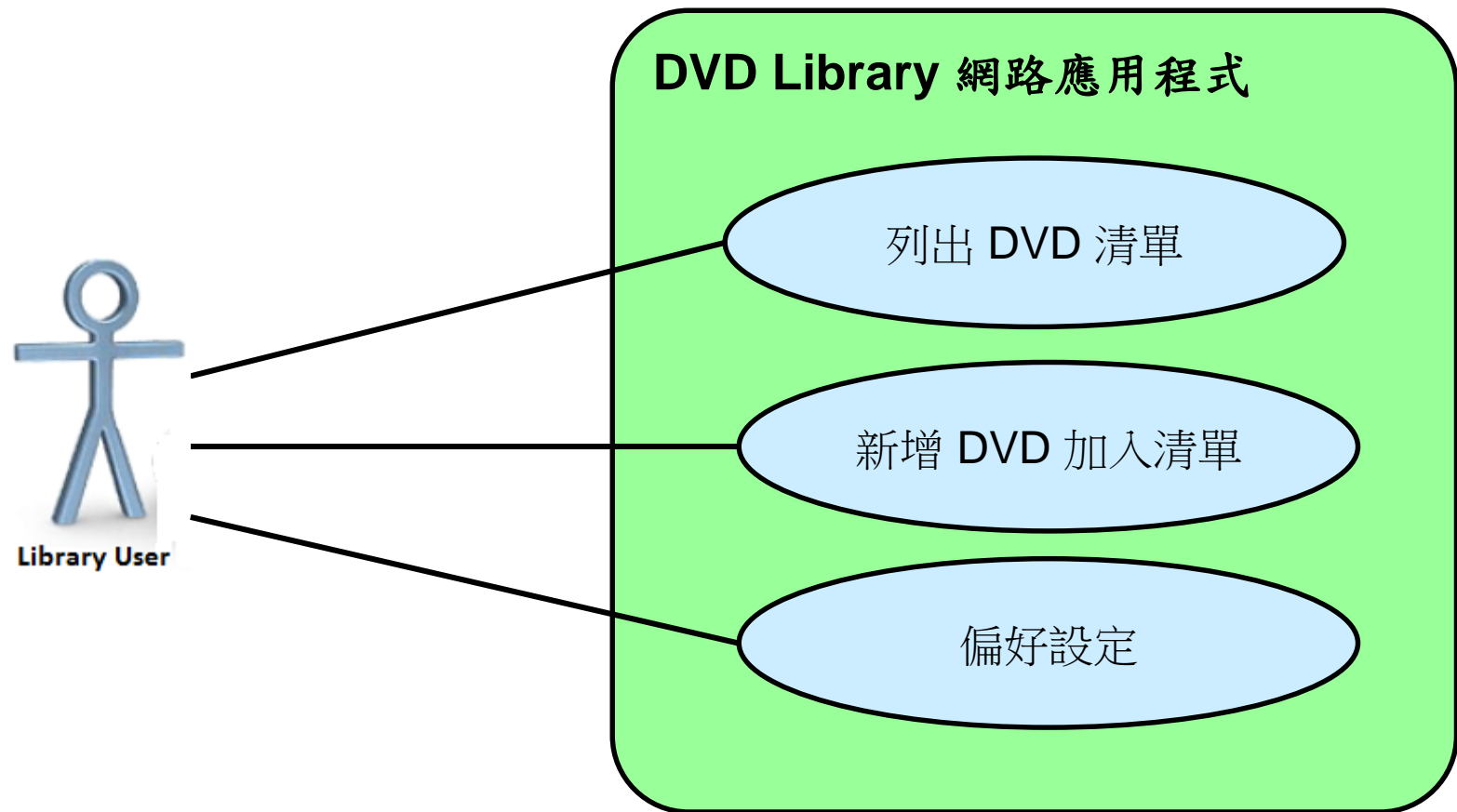


Year:

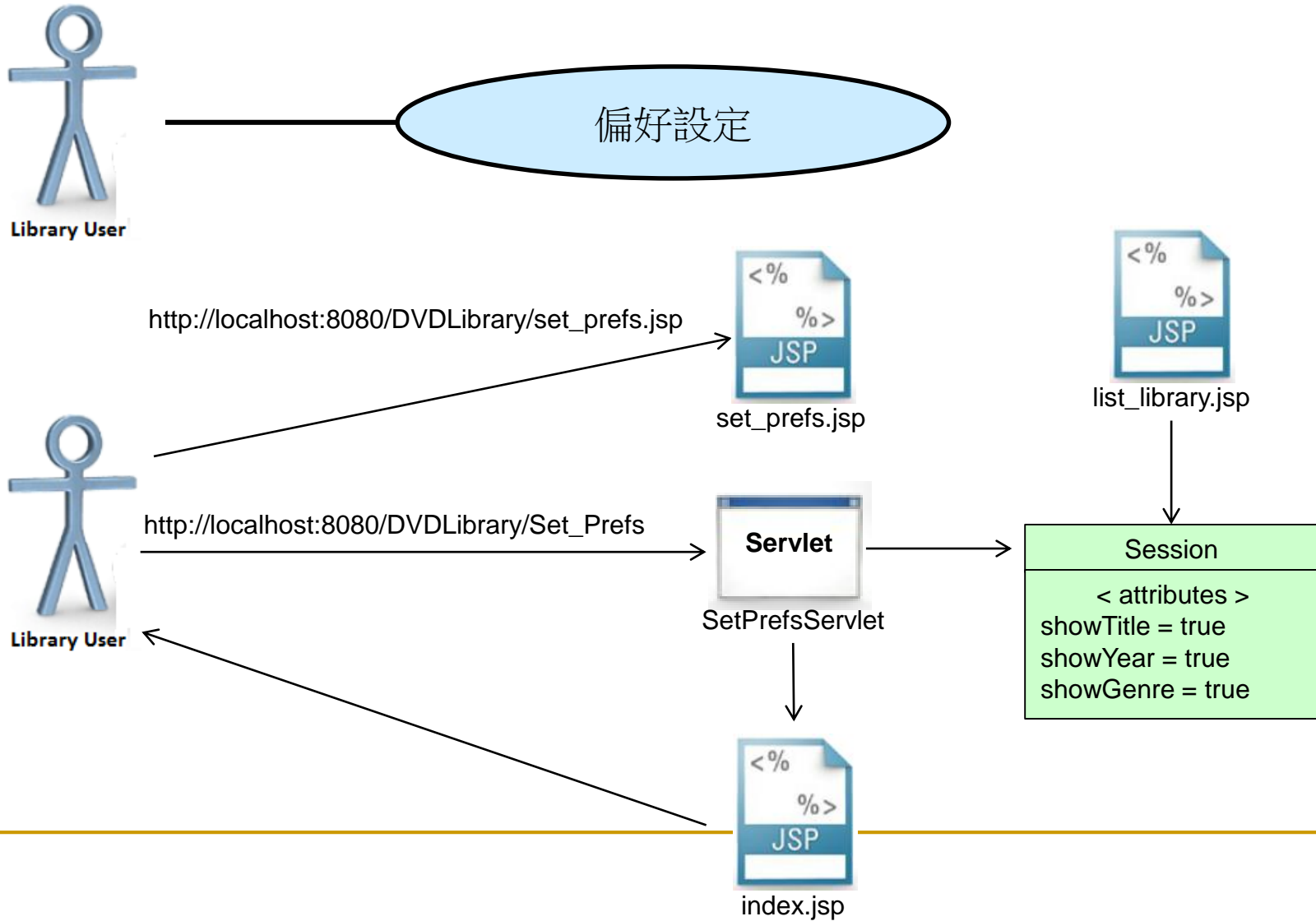
Season:



# Lab -DVDLibrary 應用程式



# Lab





# Lab

- 修改原 index.html,另存為index.jsp
  - 新增一個連結,連結至set\_prefs.jsp 頁面
    - 使用URL-Rewriting
  - 使用URL-Rewriting修改list\_library.jsp連結
  - 畫面如右上圖
- 新增 set\_prefs.jsp頁面
  - 建立一個Form表單如右下圖
  - 送出Post方法至Set\_Prefs
    - 使用URL-Rewriting
  - CheckBox 輸入欄位 show, 選項值為 showTitle, showYear, showGenre
  - 由Session中取showTitle, showYear, showGenre屬性,如果屬性存在,預先勾選 (checked)該選項
  - Submit 按鈕 Set Preferences



# Lab

- 撰寫controller.SetPrefsServlet.java
  - Servlet URL 為 Set\_Prefs
  - 取得Session物件
  - 取得請求參數show的值
    - 此參數值輸入型態為CheckBox
    - 使用getParameterValues取得一字串陣列,包含所有使用者有勾選的值
    - 如果取得的字串陣列為null,將其設為初始值{"showTitle", "showYear", "showGenre"}
  - 依客戶端傳送的請求參數陣列show,設定Session屬性
    - 移除Session中所有舊的屬性值
    - 請求參數陣列中有"showTitle"字串,在session中加入屬性showTitle
    - 請求參數陣列中有"showYear"字串,在session中加入屬性showYear
    - 請求參數陣列中有"showGenre"字串,在session中加入屬性showGenre
  - 頁面轉送至index.jsp, 使用URL-Rewriting

# Lab

- 修改 `success.jsp`
  - 回首頁改為連結 `index.jsp`
- 修改 `list_library.jsp`
  - 頁面中設定使用 `session`
  - 依 `Session` 中屬性, 設定顯示欄位如下圖
    - 由 `Session` 中取 `showTitle`, `showYear`, `showGenre` 屬性
    - 屬性存在, 顯示該欄位
    - 屬性不存在, 該欄位不顯示

