Java EE 7 Web
網路應用程式安全機制

鄭安翔

ansel_cheng@hotmail.com

課程大綱

1) **網路安全機制**
2) 網路資源授權機制
3) 身份驗證機制
4) 資料完整性及保密性機制

網路安全機制

- Authentication 身份驗證
  - 最基礎的安全機制
  - 確認使用者身份的過程
  - 常用技術為使用者名稱(username)及密碼(password)驗證機制

- Authorization 授權
  - 將網路資源根據使用者的角色(role)設定權限
  - 身份驗證與授權通常需一起使用
  - 一個網路資源集合和一個給定的用戶角色之間的映射稱為安全區域Security Domain

網路安全機制

- Data Integrity 資料完整性
  - 確保資料在傳輸過程中,沒有損壞或遭到竄改
  - 在傳輸資料中加入檢查碼或數位憑證作簽章

- Confidentiality 保密性 (Data Privacy)
  - 確保資料只有指定的接收端可以讀取
  - 典型例子為Secure Sockets Layer (SSL)加密技術,透過HTTPS(Hypertext Transfer Protocol Secure)通訊協定傳送
  - 發送端先將欲傳送的訊息加密(encrypt),只有指定接收端能將訊息解密(decrypt)

網路安全機制

- Auditing 稽核 (Access Tracking, 存取追蹤)
  - 記錄使用者在網路應用程式中所有的存取行為
  - 常見的方式為利用log方法將記錄寫入檔案或資料庫中
  - GenericServlet 類別及 ServletContext 介面提供
    - log(msg : String)
    - log(msg : String, excp : Throwable)

- Malicious Code惡意程式碼
  - 泛指一些惡意程式,例如病毒,蠕蟲,木馬程式,廣告軟體等
  - 代碼注入(Code Injection)常被用來攻擊網站應用程式
  - 安裝防火牆、防毒軟體來降低惡意程式威脅

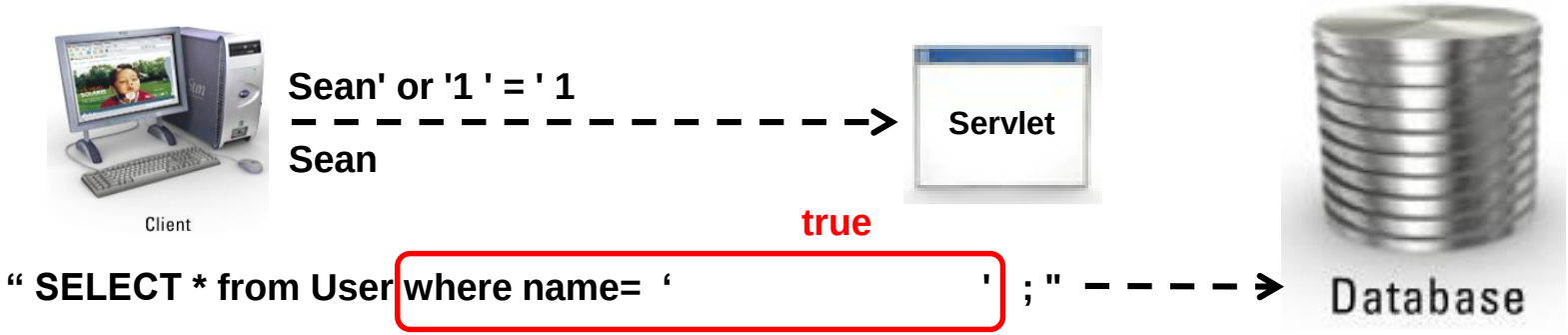
網路安全機制

- Web Attack 網路攻擊
  - 特定人士或群體對網站的攻擊
  - 常見方式包括
    - Denial of Service (阻斷服務攻擊)
    - Data Integrity Attack (資料竄改)
    - Data Privacy Attack (保密性攻擊)
      - Cracking Server Passwords(破解網站密碼)
      - Downloading Confidential Files(下載機密文件)
  - 建議解決方法

攻擊方法	建議解決方法
阻斷服務攻擊	防火牆 Firewall
資料竄改	Encrypt (SSL)
保密性攻擊	Decrypt (SSL)

SQL Injection 隱碼攻擊

- 應用程式之資料庫層的安全漏洞
  - 應用程式中使用字串串連方式組合SQL指令
  - 對使用者所輸入的資料，未限制輸入的字元數，及未對使用者輸入資料做潛在指令的檢查
  - 輸入的資料中夾帶SQL指令
  - 夾帶的指令被資料庫認為是正常的SQL指令而執行



SQL Injection

- 隱碼攻擊建議解決方法
  - 應用程式中使用參數化查詢,Prepared Statement
    - 盡量不要直接串連SQL指令
  - 過濾所有輸入字串資料
    - 檢查使用者輸入資料是否包含非預期資訊(ex:指令)
    - 只允許預期的字符

課程大綱

1) 網路安全機制
2) **網路資源授權機制**
   - **定義需要保護的網路資源集合**
   - **定義安全性角色**
   - **網路資源集合對應可存取的安全性角色**
   - 對安全性角色設定對應之使用者帳號
3) 身份驗證機制
4) 資料完整性及保密性機制

網路資源授權機制

- Java Servlet 標準提供宣告式授權機制
  - 使用web.xml檔,宣告哪些使用者可以使用哪些資源
- Web.xml 中宣告式授權相關宣告
  - 安全性角色 security role 標籤

標籤名稱	出現次數	說明
<security-role>	*	安全性角色
<description>	?	角色描述
<role-name>	*	角色名稱

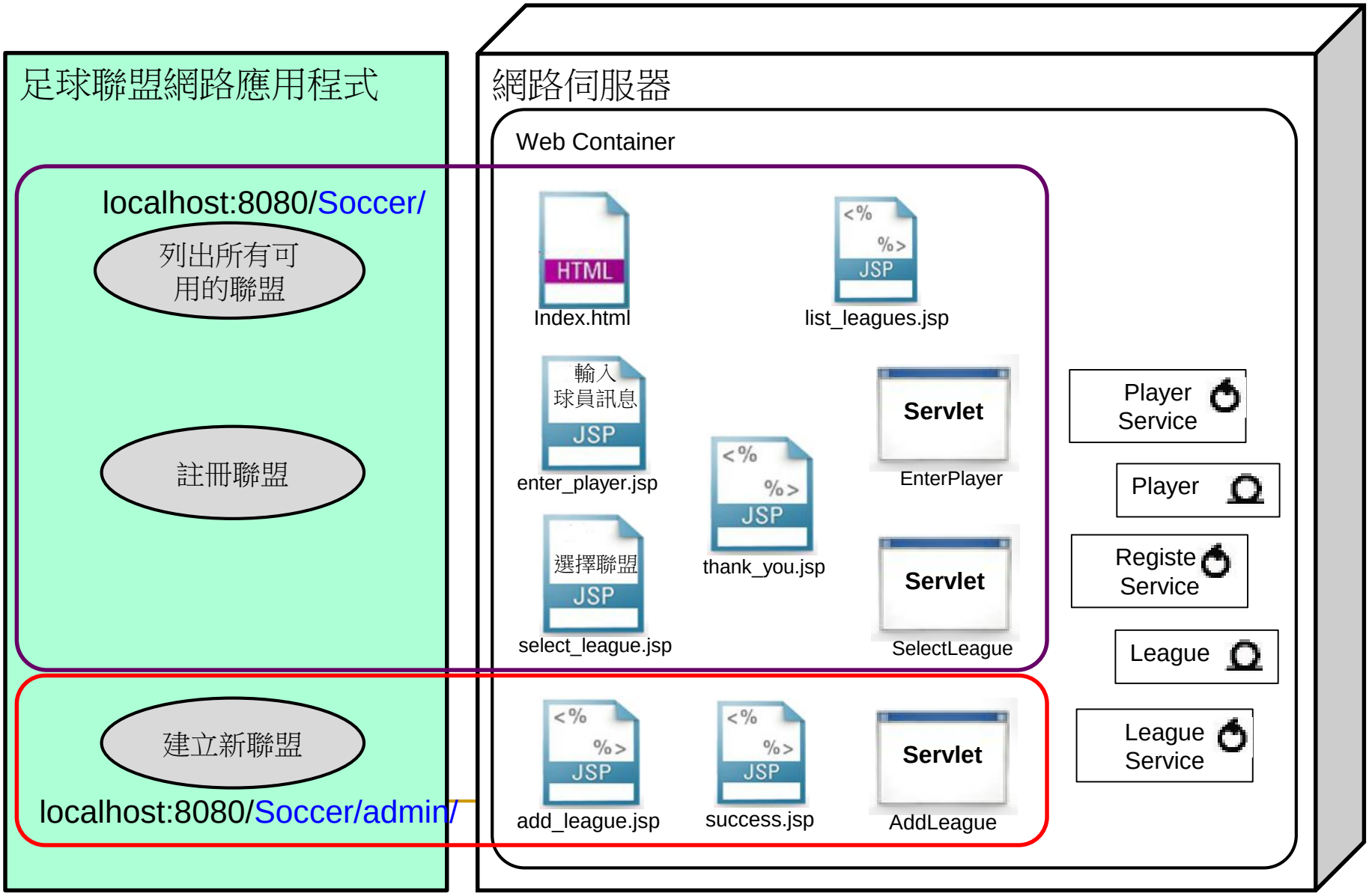
  - 安全性限制條件 security constraint 標籤
    - 網路資源集合 web-resource-collection 標籤
    - 角色限制設定auth-constraint 標籤
      - 定義資源集合可存取之使用者角色(role)

標籤名稱	出現次數	說明
<security-constraint>	*	安全性限制
<display-name>	?	安全資源名稱
<web-resource-collection>	+	受保護的網路資源集合
<web-resource-name>	1	網路資源集合名稱
<description>	?	資源描述
<url-pattern>	*	資源集合所在的URL
<http-method>	*	受保護的HTTP方法 通過授權者才可使用該方法存取該資源 未設定此標籤則所有連線方法皆不可使用
<auth-constraint >	?	角色限制設定
<description>	?	限制描述
<role-name>	*	可存取資源的安全性角色 對應<security-role>標籤所定義的角色
<user-data-constraint>	?	傳輸限制設定
<description>	?	傳輸限制描述
<transport-guarantee>	1	通訊層設定的保護方式 NONE:不保護隱私性及完整性 INTEGRAL:確保資料完整性 CONFIDENTIAL:確保資料隱私性及完整性

宣告式授權機制

- 宣告式授權實施步驟
  1) 定義需要保護的網路資源集合
  2) 定義安全性角色
  3) 網路資源集合對應可存取的安全性角色
  4) 對安全性角色設定對應之使用者帳號

定義需要保護的網路資源集合



web.xml中定義需保護的資源集合

```
<web-app>
    <display-name>  Web application 名稱.  </display-name>
    <description>  Web application 說明.  </description>
    <servlet>……
    <servlet-mapping>….
    <security-constraint>
        <web-resource-collection>
            <web-resource-name>資源集合名稱 </web-resource-name>
            <description>資源集合描述 </description>
            <url-pattern>資源集合所在之URL </url-pattern>
            <http-method> HTTP方法 </http-method> *
        </web-resource-collection>
        <auth-constraint>
            <role-name>安全性角色名稱 </role-name>
         </auth-constraint>
   </security-constraint>
    ……
</web-app>
```

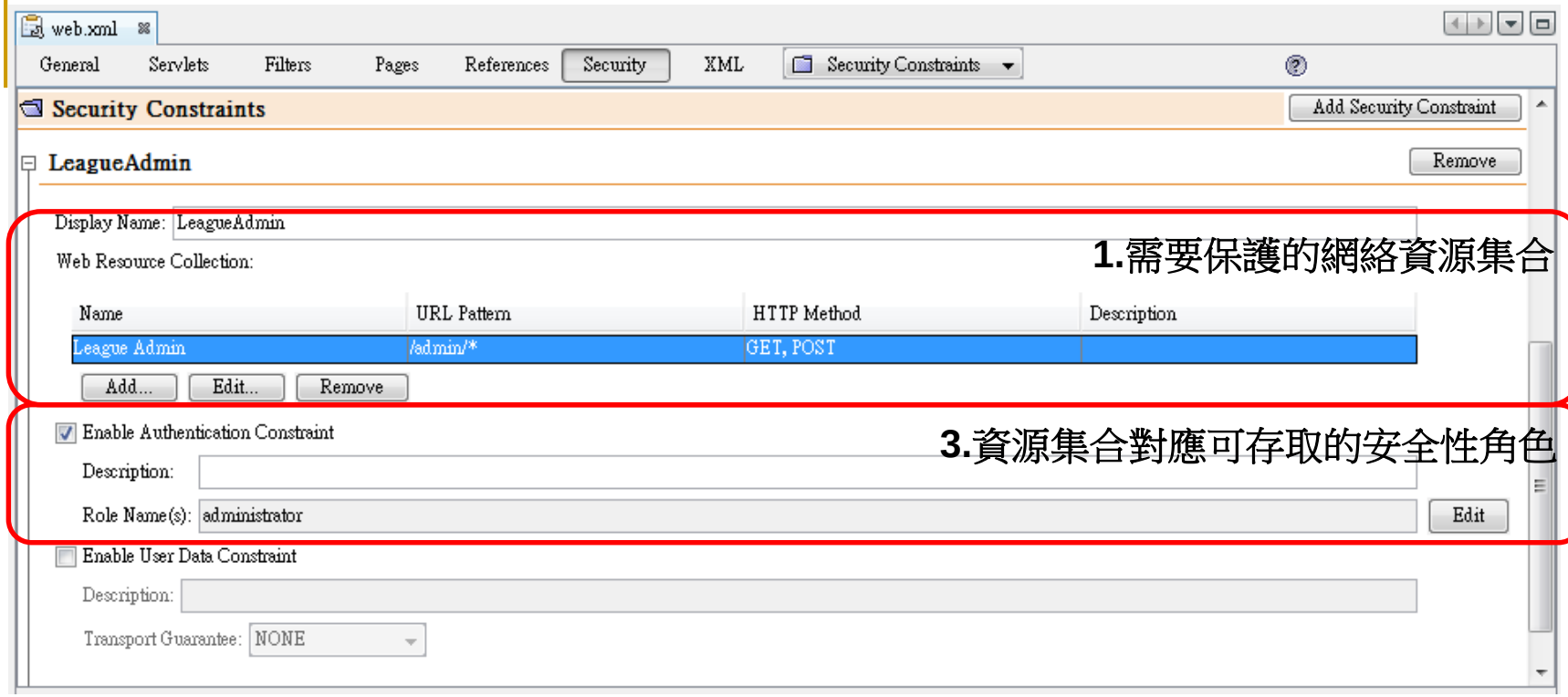
定義安全性角色

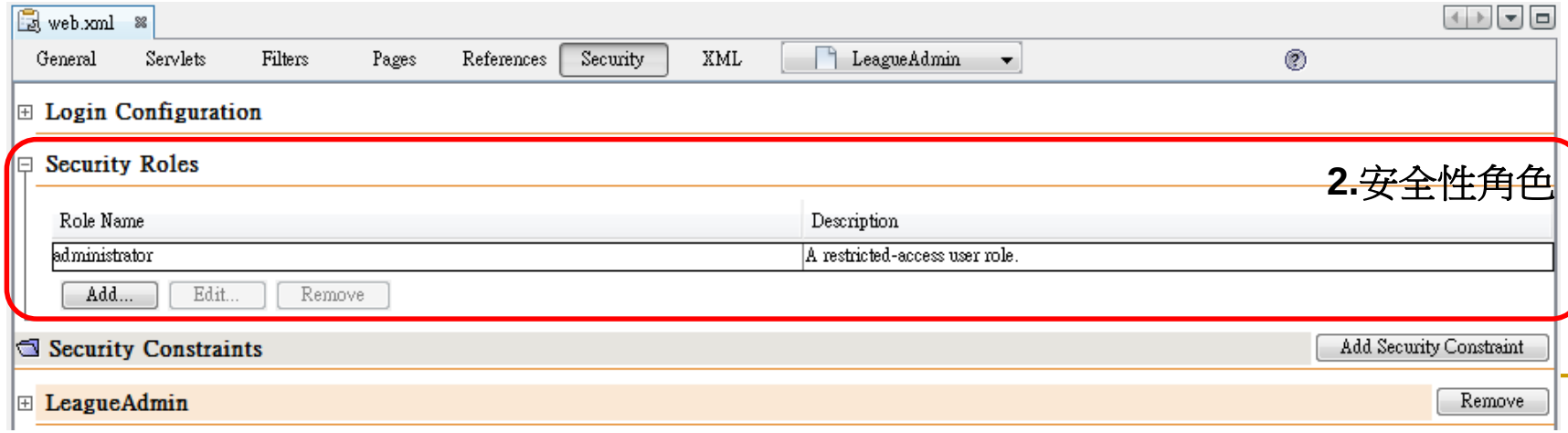
- web.xml中定義安全性角色

```
<web-app>
   <display-name>  Web application 名稱.  </display-name>
   <description>  Web application 說明.  </description>
   <servlet>……
   <servlet-mapping>….
   <security-constraint>
       ….
       <auth-constraint>
          <role-name>安全性角色名稱 </role-name>
       </auth-constraint>
       ….
   </security-constraint>
   <security-role>
         <description>安全性角色說明 </description>
         <role-name>安全性角色名稱 </role-name>
    </security-role>
   ….
</web-app>
```

資源集合對應可存取的安全性角色

```
<web-app>
    <display-name>  Web application 名稱.  </display-name>
    <servlet>……
    <servlet-mapping>….
    <security-constraint>
        <web-resource-collection>
            <web-resource-name>資源集合名稱 </web-resource-name>
            <description>資源集合描述 </description>
            <url-pattern>資源集合所在之URL </url-pattern>
            <http-method> HTTP方法 </http-method> *
        </web-resource-collection>
        <auth-constraint>
            <role-name>安全性角色名稱 </role-name>
        </auth-constraint>
    </security-constraint>
    <security-role>
            <description>安全性角色說明 </description>
            <role-name>安全性角色名稱 </role-name>
     </security-role>
     ……
</web-app>
```


web.xml
General
Servlets
Filters
Pages
References
Security
XML
Security Constraints
Security Constraints
Add Security Constraint
LeagueAdmin
Remove
Display Name: LeagueAdmin
Web Resource Collection:
Name
URL Pattern
HTTP Method
Description
League Admin
/admin/*
GET, POST
Add...
Edit...
Remove
Enable Authentication Constraint
Description:
Role Name(s): administrator
Edit
Enable User Data Constraint
Description:
Transport Guarantee: NONE
1.需要保護的網絡資源集合
3.資源集合對應可存取的安全性角色


web.xml
General
Servlets
Filters
Pages
References
Security
XML
LeagueAdmin
Login Configuration
Security Roles
Role Name
Description
administrator
A restricted-access user role.
Add...
Edit...
Remove
Security Constraints
Add Security Constraint
LeagueAdmin
Remove
2.安全性角色

課程大綱

1) 網路安全機制
2) **網路資源授權機制**
   - 定義需要保護的網路資源集合
   - 定義安全性角色
   - 網路資源集合對應可存取的安全性角色
   - **對安全性角色設定對應之使用者帳號**
3) 身份驗證機制
4) 資料完整性及保密性機制

對安全性角色設定對應使用者帳號

- 安全性領域 Security Realm
  - 定義使用者帳號與安全性角色的對應關係
  - 通常也作帳號與密碼的驗證(Authentication)
  - 網路元件容器均需實作安全性領域元件,實作方法包括
    - 純文字檔
    - 資料庫
    - LDAP (Lightweight Directory Access Protocol)
    - Java Authentication and Authorization Service
    - Network Information System (NIS)
  - 網路元件容器通常可同時支援多種安全領域實作

Tomcat 純文字檔安全性領域實作

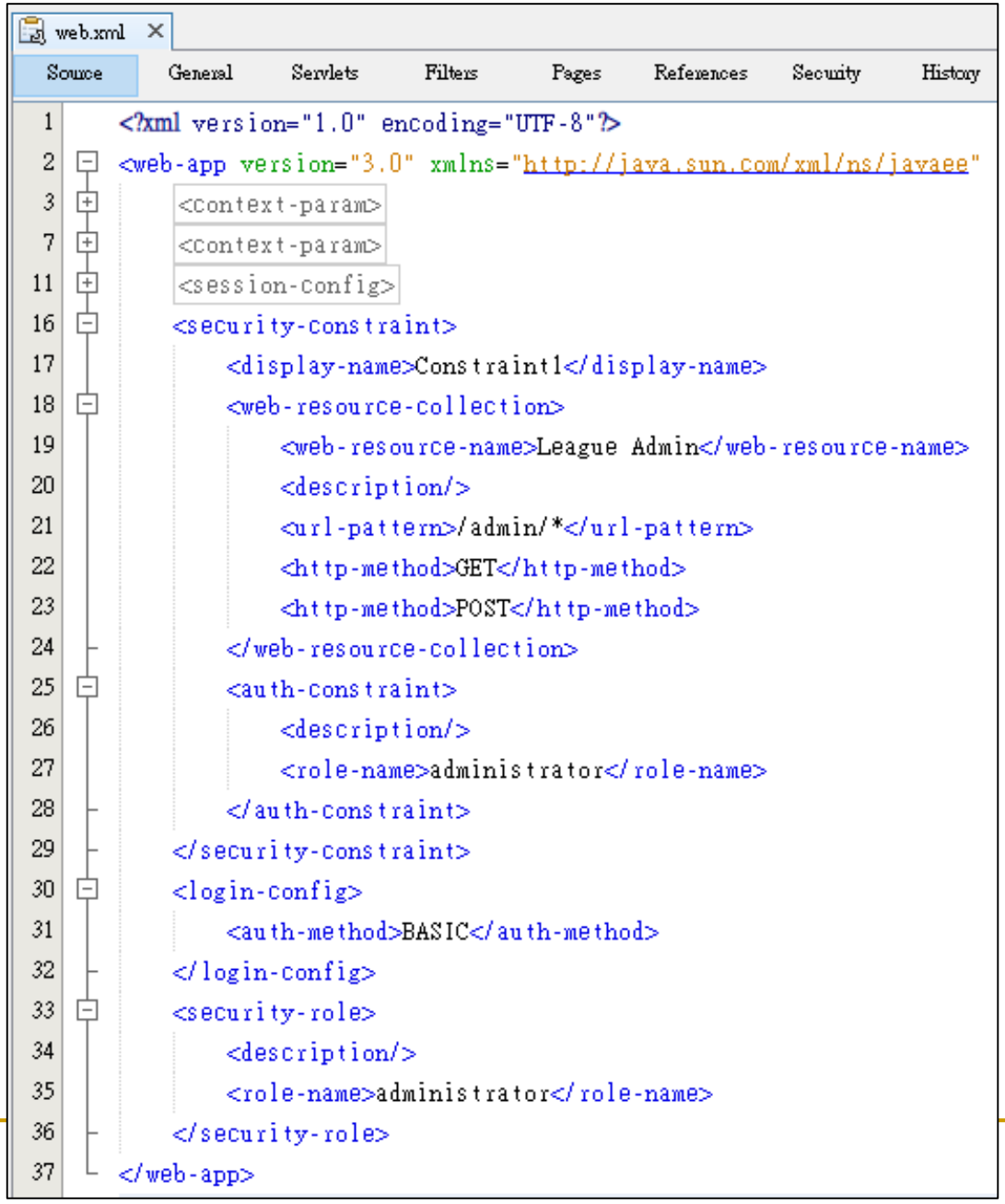
- <TOMCAT_HOME>\conf\tomcat-users.xml
  - 定義使用者ID, Password 及所屬之security role

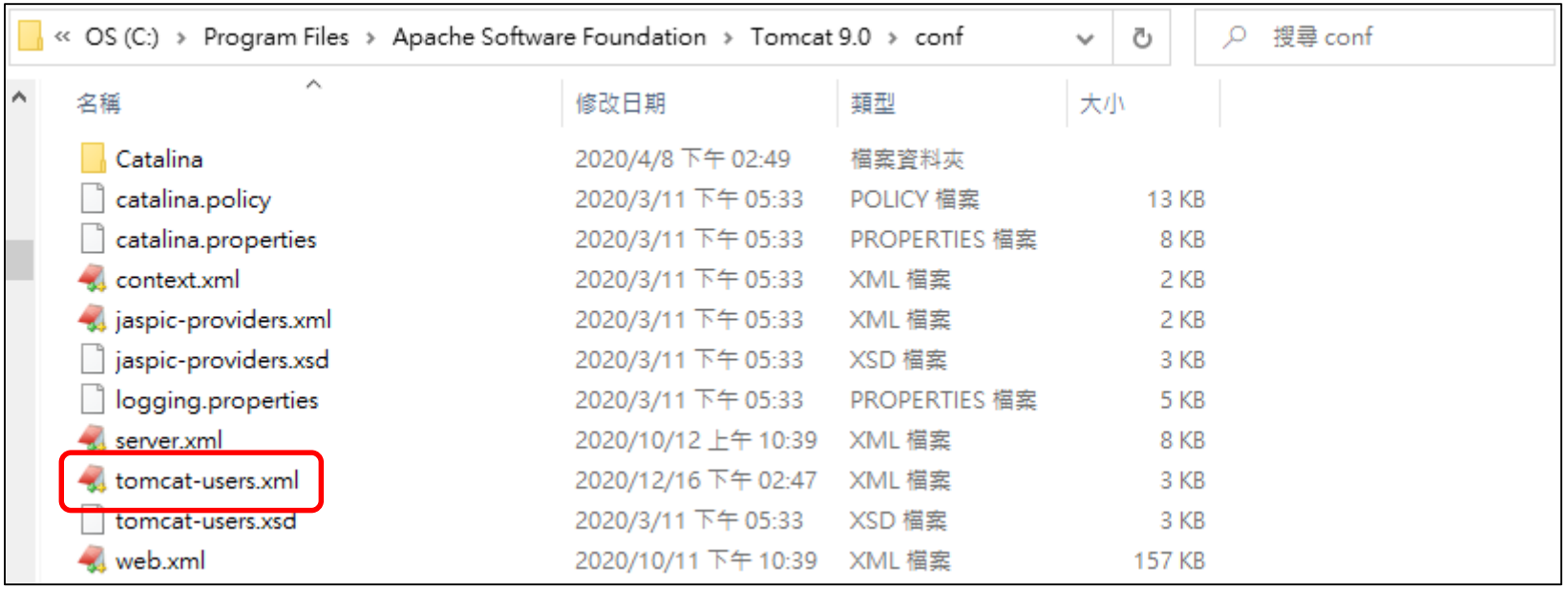
<role rolename =“*安全性角色*“/>

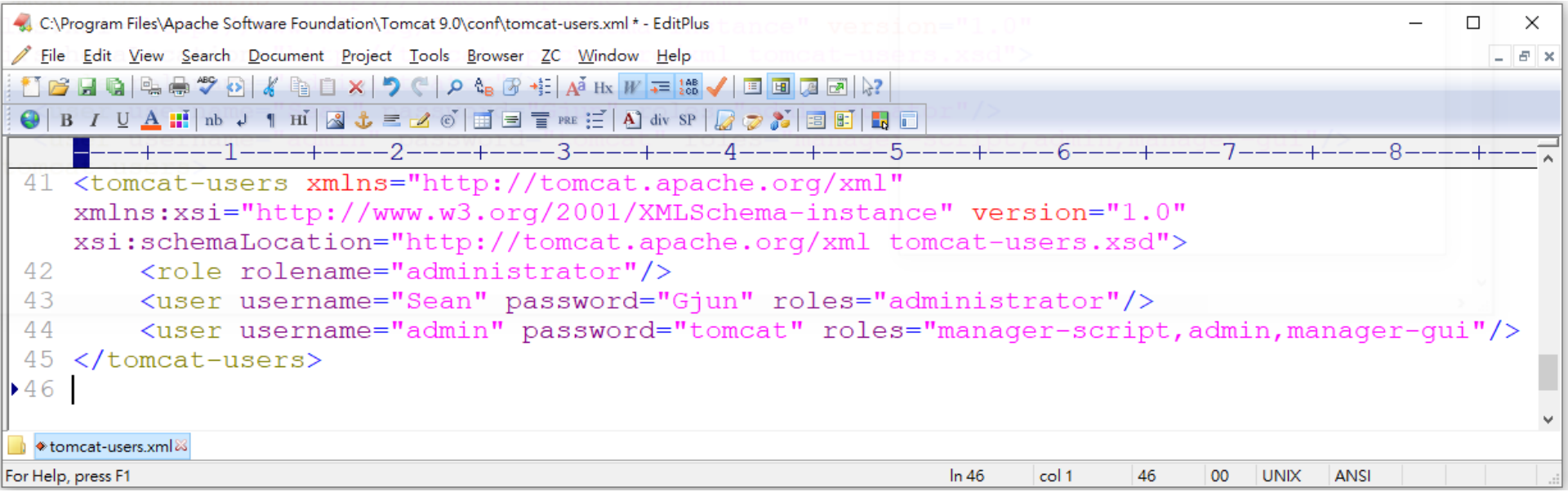
<user usermame=“*使用者名稱*“ password=“*密碼*“ roles=“*安全性角色*“/>

tomcat-users.xml

```
<tomcat-users>
   <role rolename="administrator"/>
   <user uesrname="Sean" password="Gjun" roles="administrator" />
</tomcat-users>
```


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"
3 <context-param>
7 <context-param>
11 <session-config>
16 <security-constraint>
17 <display-name>Constraint1</display-name>
18 <web-resource-collection>
19 <web-resource-name>League Admin</web-resource-name>
20 <description/>
21 <url-pattern>/admin/*</url-pattern>
22 <http-method>GET</http-method>
23 <http-method>POST</http-method>
24 </web-resource-collection>
25 <auth-constraint>
26 <description/>
27 <role-name>administrator</role-name>
28 </auth-constraint>
29 </security-constraint>
30 <login-config>
31 <auth-method>BASIC</auth-method>
32 </login-config>
33 <security-role>
34 <description/>
35 <role-name>administrator</role-name>
36 </security-role>
37 </web-app>


« OS (C:) › Program Files › Apache Software Foundation › Tomcat 9.0 › conf
搜尋 conf
名稱
修改日期
類型
大小
Catalina 2020/4/8 下午 02:49 檔案資料夾
catalina.policy 2020/3/11 下午 05:33 POLICY 檔案 13 KB
catalina.properties 2020/3/11 下午 05:33 PROPERTIES 檔案 8 KB
context.xml 2020/3/11 下午 05:33 XML 檔案 2 KB
jaspic-providers.xml 2020/3/11 下午 05:33 XML 檔案 2 KB
jaspic-providers.xsd 2020/3/11 下午 05:33 XSD 檔案 3 KB
logging.properties 2020/3/11 下午 05:33 PROPERTIES 檔案 5 KB
server.xml 2020/10/12 上午 10:39 XML 檔案 8 KB
tomcat-users.xml 2020/12/16 下午 02:47 XML 檔案 3 KB
tomcat-users.xsd 2020/3/11 下午 05:33 XSD 檔案 3 KB
web.xml 2020/10/11 下午 10:39 XML 檔案 157 KB


C:\Program Files\Apache Software Foundation\Tomcat 9.0\conf\tomcat-users.xml * - EditPlus
File Edit View Search Document Project Tools Browser ZC Window Help
41 <tomcat-users xmlns="http://tomcat.apache.org/xml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0"
xsi:schemaLocation="http://tomcat.apache.org/xml tomcat-users.xsd">
42 <role rolename="administrator"/>
43 <user username="Sean" password="Gjun" roles="administrator"/>
44 <user username="admin" password="tomcat" roles="manager-script,admin,manager-gui"/>
45 </tomcat-users>
46
tomcat-users.xml
For Help, press F1
ln 46 col 1 46 00 UNIX ANSI

課程大綱

1) 網路安全機制
2) 網路資源授權機制
3) **Servlet身份驗證機制**
   - **基礎驗證 HTTP Basic Authentication**
   - **摘要式驗證 HTTP Digest Authentication**
   - **HTTPS 協定**
   - **表單式驗證 FORM-Based Authentication**
   - **取得使用者登入身份**
4) 資料完整性及保密性機制

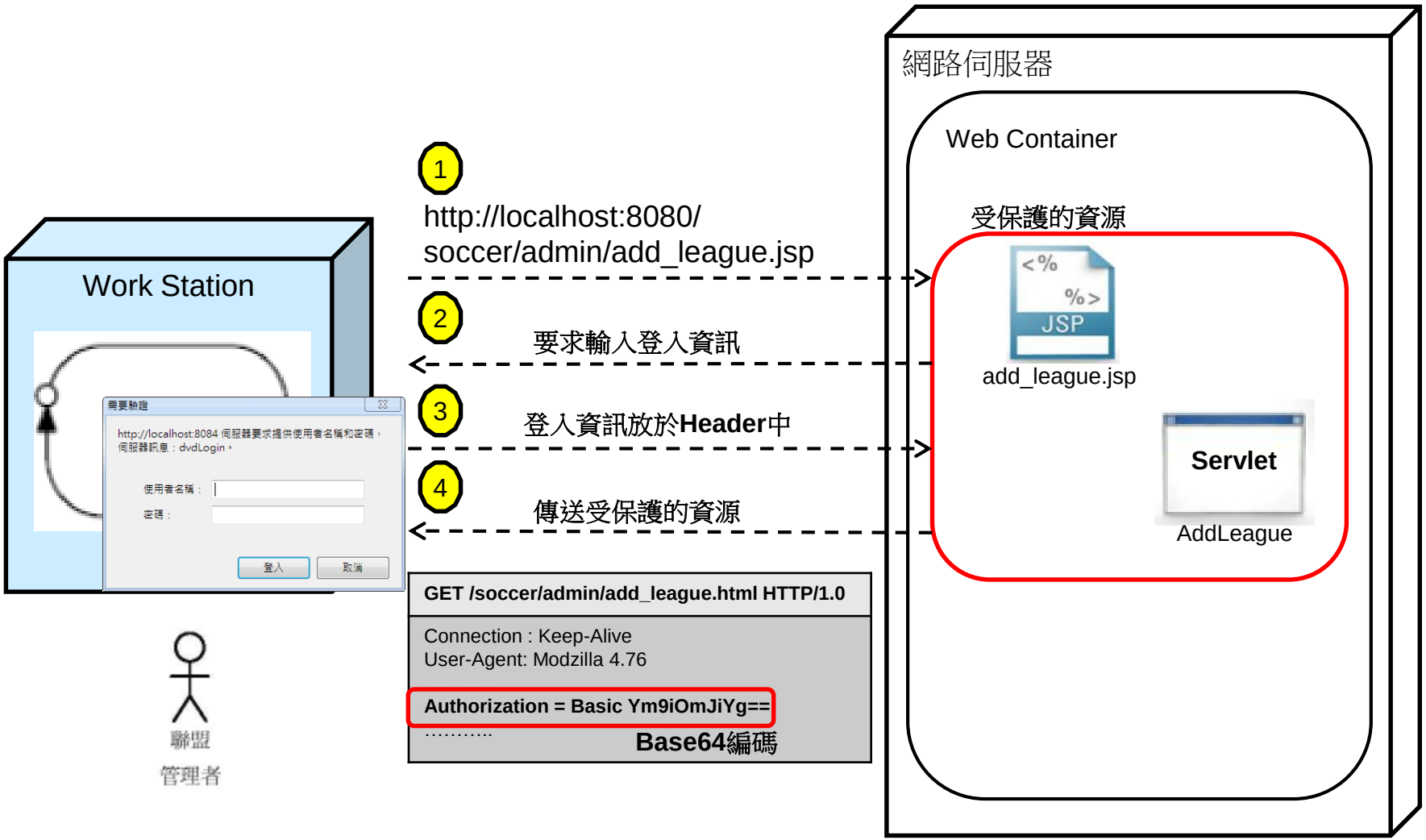
Servlet 支援的身份驗證機制

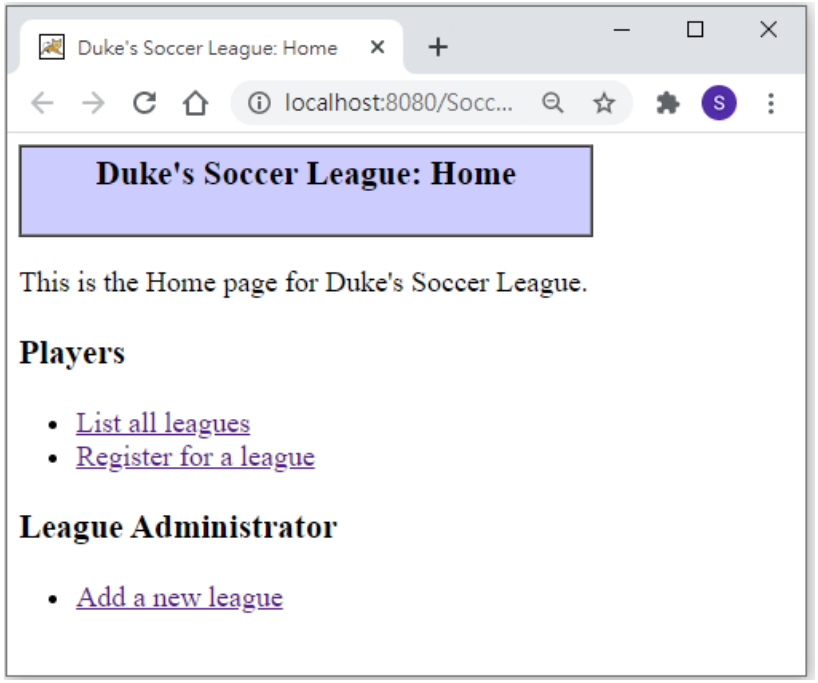
- Java Servlet 標準支援四種身份驗證機制
  - 基礎驗證 HTTP Basic Authentication
  - 摘要式驗證 HTTP Digest Authentication
  - HTTPS 協定
  - 表單式驗證 FORM-Based Authentication

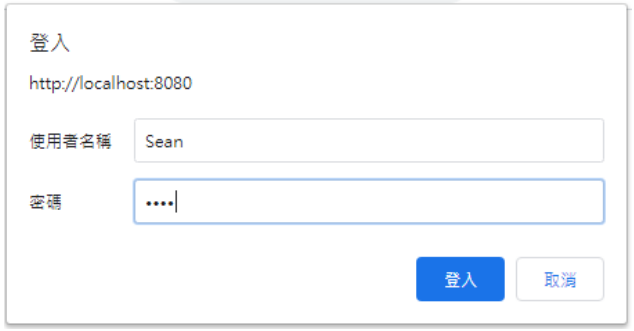
基礎驗證 Basic Authentication

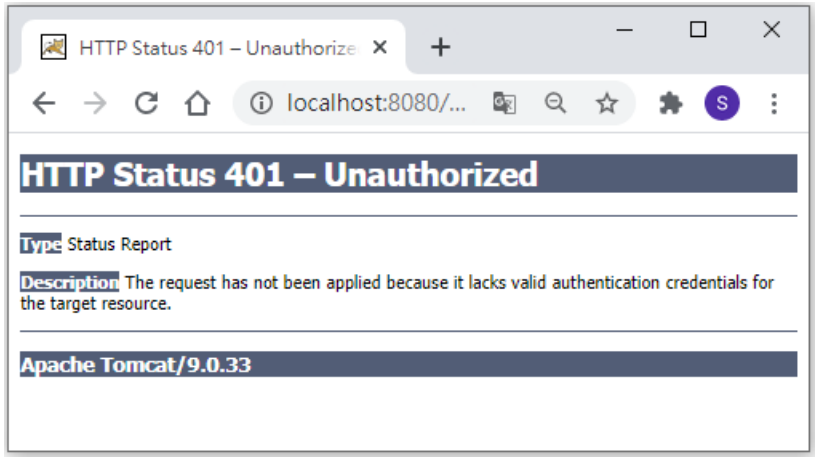
- 基礎驗證 HTTP Basic Authentication
  - 定義於HTTP 1.0規格中
  - 傳輸時使用Base64編碼
    - Base64是一種公開的演算法
    - 最簡單的驗證機制
    - 安全性較低
  - 帳號密碼輸入使用瀏覽器預設的對話框

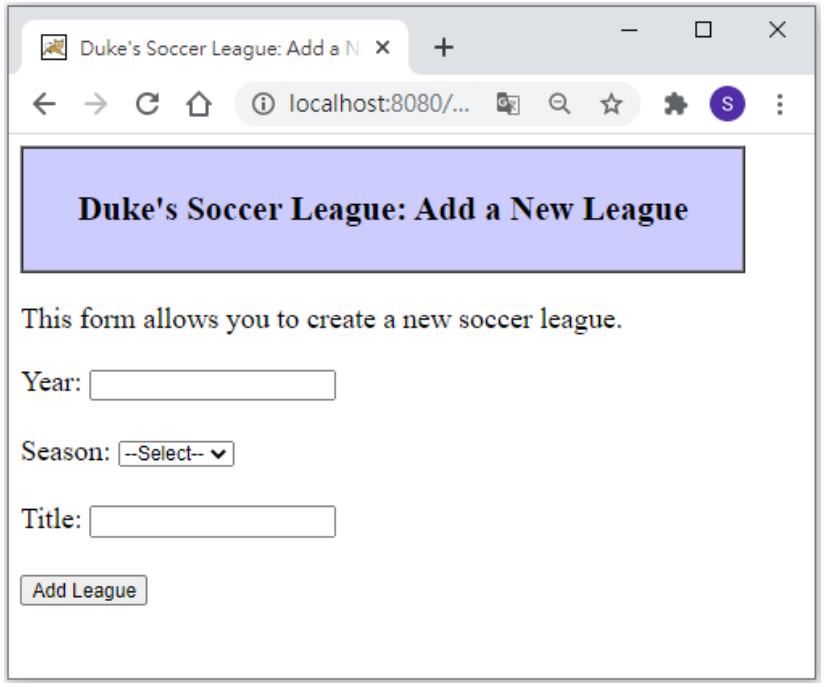
Basic Authentication




Duke's Soccer League: Home
localhost:8080/Socc...
Duke's Soccer League: Home
This is the Home page for Duke's Soccer League.
Players
List all leagues
Register for a league
League Administrator
Add a new league


登入
http://localhost:8080
使用者名稱
Sean
密碼
登入
取消


HTTP Status 401 – Unauthorize
localhost:8080/...
HTTP Status 401 – Unauthorized
Type Status Report
Description The request has not been applied because it lacks valid authentication credentials for the target resource.
Apache Tomcat/9.0.33


Duke's Soccer League: Add a N
localhost:8080/...
Duke's Soccer League: Add a New League
This form allows you to create a new soccer league.
Year:
Season: --Select--
Title:
Add League

Lab 1

- 修改 ContactInfo 網路應用專案
  - 不包含Filter機制 (與第九章練習完成結果相同)
  - 以內建安全機制保護要求使用者登入
- Tomcat 中設定使用者帳號及角色
  - <TOMCAT_HOME>\conf\tomcat-users.xml
  - 新增腳色
  - 新增使用者帳號並指定角色

Lab 1

- 部署描述檔設定安全機制
  - 新增部署描述檔(web.xml)
  - 設定身份驗證模式為基礎驗證
    - web.xml中選擇 security標籤
    - 展開Login Configuration, 選擇 basic
  - 宣告安全性腳色
    - 展開Security Roles
    - 新增安全性角色user

Lab 1

- 宣告安全性限制條件
  - web.xml security標籤中Security Constraints 資料夾
  - 新增Security Constraint
    - Display Name 欄位輸入 Constraint1
  - 新增網路資源集合
    - Web Resource Collections 按新增按鈕
    - Resource Name 欄位輸入 MyResource
    - URL Patterns 欄位輸入 /SelectCustomer
  - 設定資源集合可操作之安全性角色
    - 勾選 Enable Authentication Constraints
    - 編輯 Role Name, 輸入 user
- 測試、執行

摘要式驗證 Digest Authentication

- 摘要式驗證 HTTP Digest Authentication
  - 驗證流程與基礎驗證相同
  - 傳輸時使用MD5編碼
    - 安全性優於Base64編碼
  - 帳號密碼輸入使用瀏覽器預設的對話框
    - 瀏覽器不一定支援
    - IE5.0以上版本支援
  - JavaEE規格書中並未規定一定要提供此驗證方法
    - 網路元件容器可能不支援

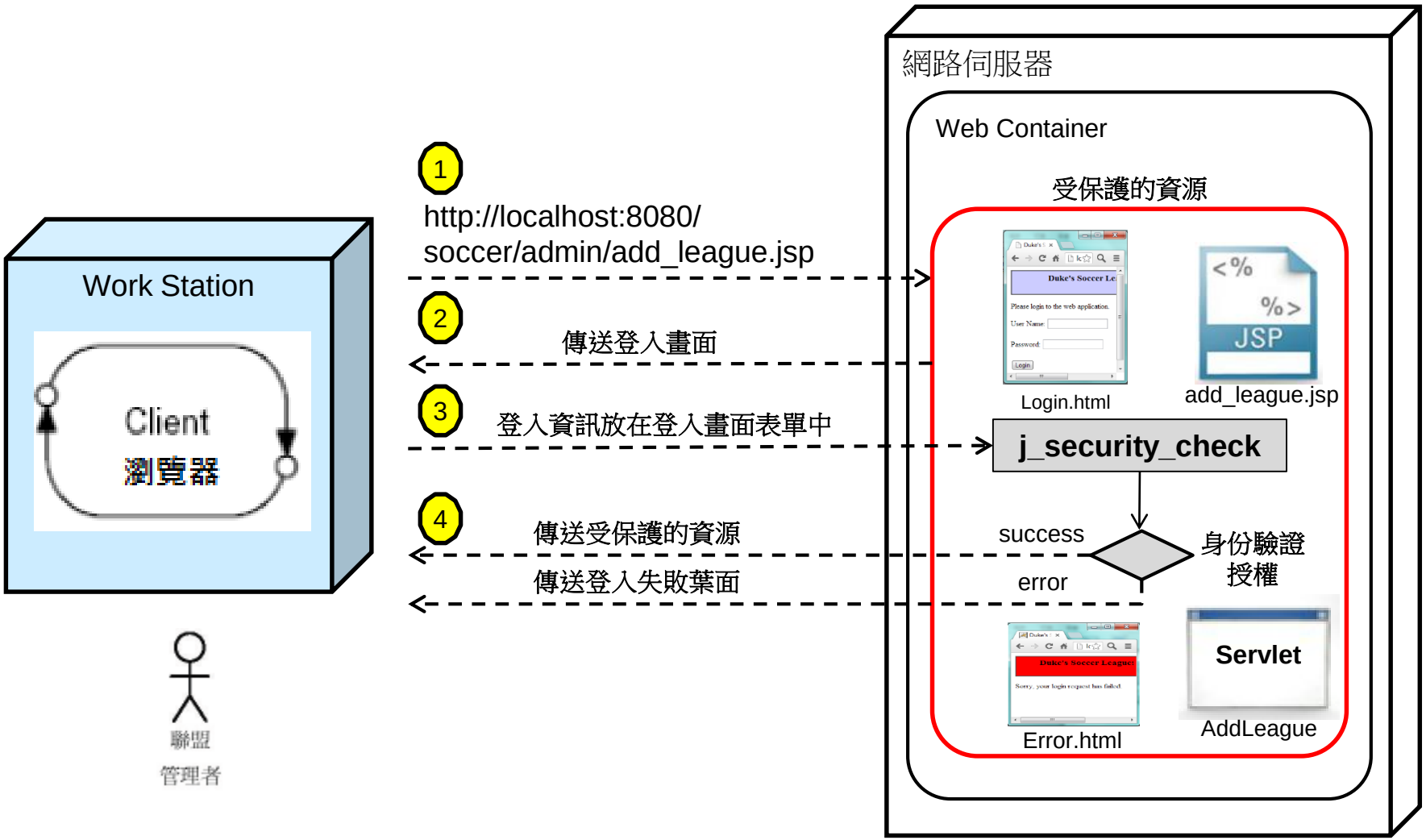
HTTPS 驗證

- 超文字傳輸安全協定 (Hypertext Transfer Protocol Secure)
  - 建構在SSL(Secure Socket Layer)上的HTTP通訊
  - 利用密鑰演算法在網際網路上提供端點身份認證與通訊保密
  - 公鑰基礎設施Public Key Infrastructure(PKI)
  - SSL是目前較安全的網路加密技術, 幾乎所有瀏覽器都支援
  - SSL並非免費,需透過認證授權機構(Certificate Authority)提供密鑰
    - 電子簽名證書相當昂貴

表單式驗證 FORM-Based

- 表單式驗證 FORM-Based Authentication
  - 驗證流程與架構與基礎驗證相同
  - 傳輸時使用Base64編碼
    - 安全性較低
    - 也可搭配HTTPS
  - 帳號密碼輸入使用客製化HTML表單
    - 所有瀏覽器皆支援
  - 實作時,可搭配Session管理或JavaEE提供之Form-Based 安全性設定指令

FORM-Based Authentication



宣告式身份驗證

- 網路元件容器中建立使用者帳號,密碼,安全性角色
- web.xml中設定宣告式身份驗證標籤

標籤名稱	出現次數	說明
<login-config>	?	登入畫面設定
<auth-method>	?	身份驗證方式：BASIC、DIGEST、CLIENT_CERT、FORM
<realm-name>	?	設定領域名稱
<form-login-config>	?	FORM驗證時頁面URL
<form-login-page>	?	登入頁面URL
<form-login-error>	?	錯誤頁面URL

宣告式身份驗證

- web.xml中設定驗證方法
  - BASIC

```
<login-config>
    <auth-method>BASIC</auth-method>
</login-config>
```

  - DIGEST

```
<login-config>
    <auth-method>DIGEST</auth-method>
</login-config>
```

宣告式身份驗證

- HTTPS

```
<login-config>
    <auth-method>CLIENT_CERT</auth-method>
</login-config>
```

- FORM-Based

```
<login-config>
    <auth-method>FORM</auth-method>
    <form-login-config>
        <form-login-page>登入畫面URL</form-login-page>
        <form-error-page>登入失敗畫面URL</form-error-page>
    </form-login-config>
</login-config>
```

撰寫自訂登入及登入失敗頁面

- JavaEE容器內建之安全性特殊屬性
  - 網路元件容器處理驗證工作,開發者不需自行實作Session管理
  - Form ACTION

    <FORM ACTION='j_security_check' METHOD='POST'>
  - User name

    <INPUT TYPE='text' NAME='j_username' SIZE='50'>
  - Password

    <INPUT TYPE='password' NAME='j_password' SIZE='50'>

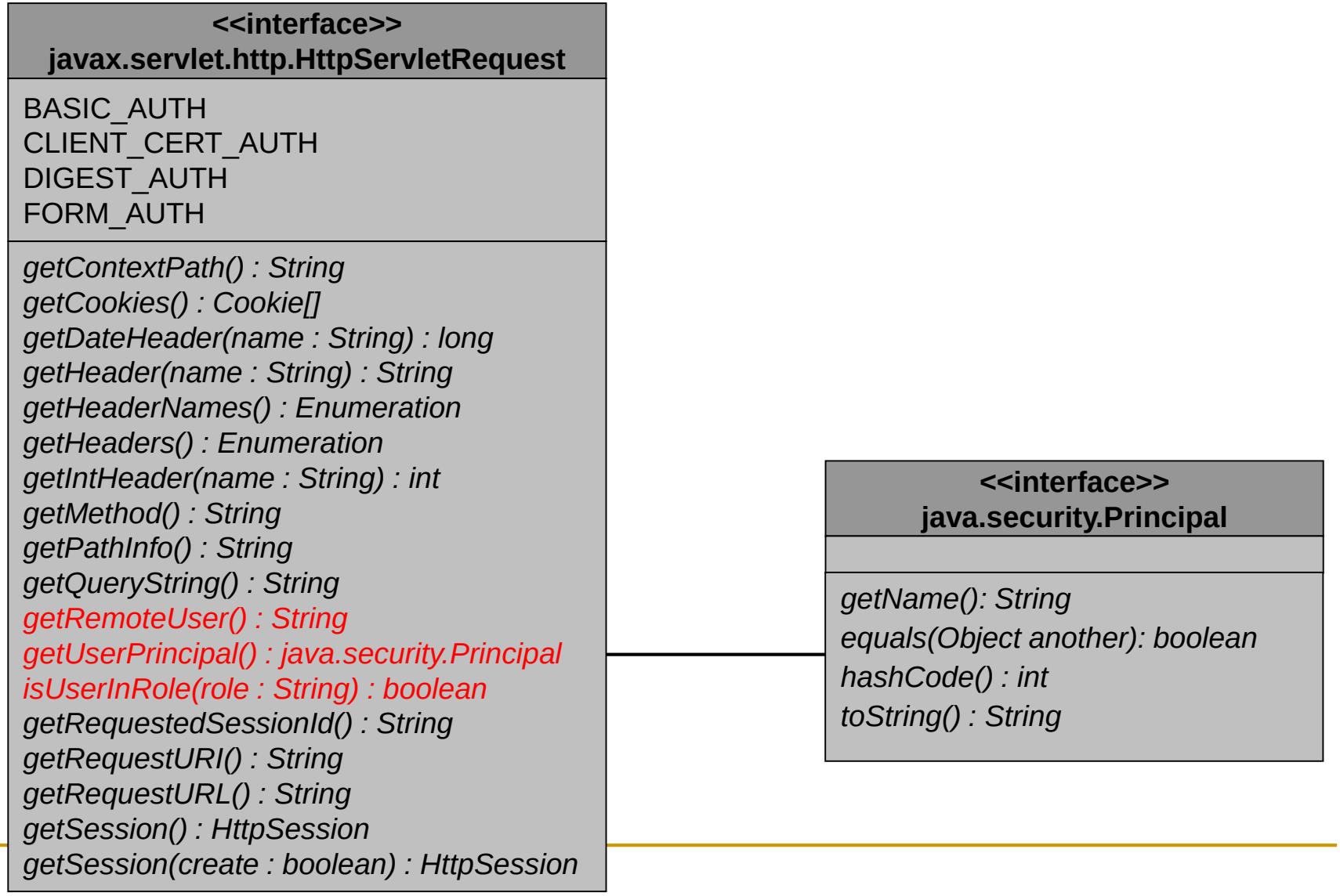
/login/form.html

```
01 <html>
02 <head>
03    <title>Duke's Soccer League: Login</title>
04 </head>
05 <body bgcolor='white'>
06 <!-- Page Heading -->
07 <table border='1' cellpadding='5' cellspacing='0' width='400'>
08    <tr bgcolor='#CCCCFF' align='center' valign='center' height='20'>
09       <td><h3>Duke's Soccer League: Login</h3></td>
10    </tr>
11 </table>
12 <p>
13    Please login to the web application.
14 </p>
15 <form action='j_security_check' method='POST'>
16    User Name: <input type='text' name='j_username' size='16'/> <br><br>
17    Password: <input type='password' name='j_password' size='16'/><br><br>
18    <input type='submit' value='Login' />
19 </form>
20 </body>
21 </html>
```

/login/error.html

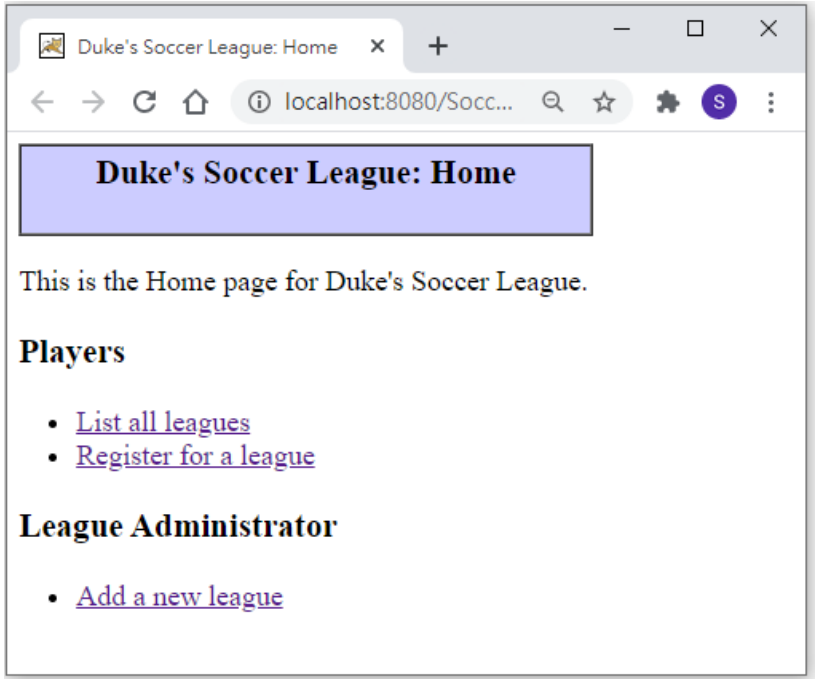
```
01  <html>
02  <head>
03  <title>Duke's Soccer League: Login Failure</title>
04  </head>
05  <body bgcolor='white'>
06  <!-- Page Heading -->
07  <table border='1' cellpadding='5' cellspacing='0' width='400'>
08      <tr bgcolor='red' align='center' valign='center' height='20'>
09          <td><h3>Duke's Soccer League: Login Failure</h3></td>
10      </tr>
11  </table>
12  <p>
13      Sorry, your login request has failed.
14  </p>
15  </body>
16  </html>
```

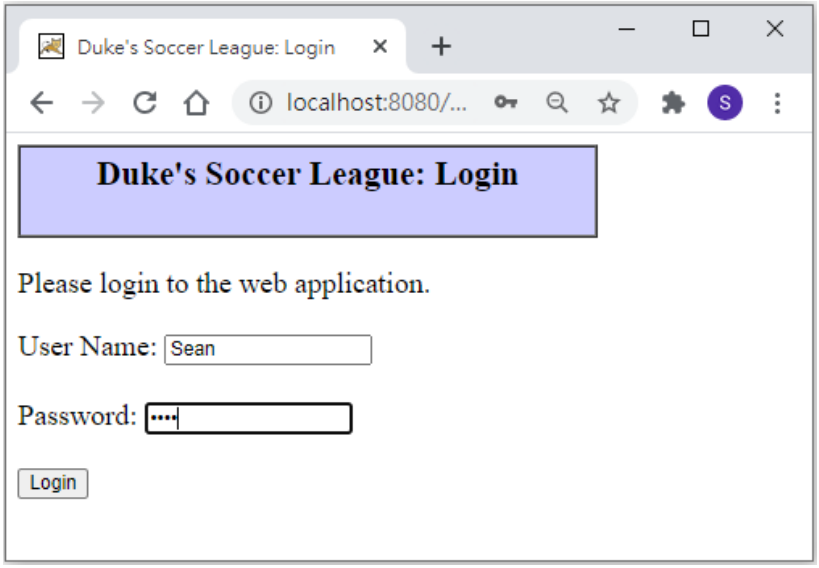
取得使用者登入身份

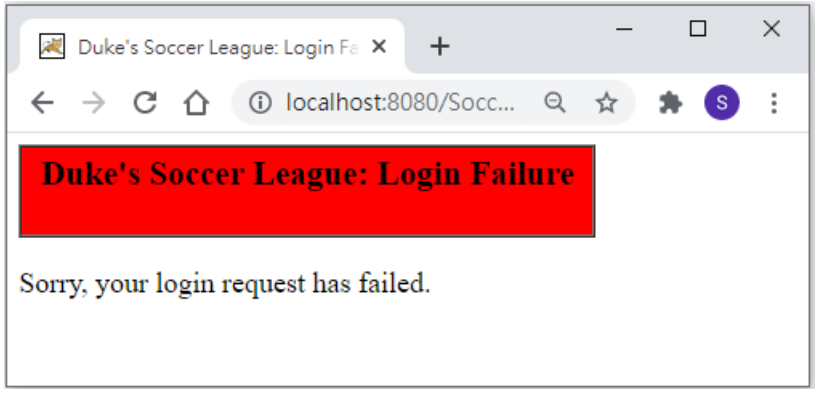

<<interface>>
javax.servlet.http.HttpServletRequest
BASIC_AUTH
CLIENT_CERT_AUTH
DIGEST_AUTH
FORM_AUTH
getContextPath() : String
getCookies() : Cookie[]
getDateHeader(name : String) : long
getHeader(name : String) : String
getHeaderNames() : Enumeration
getHeaders() : Enumeration
getIntHeader(name : String) : int
getMethod() : String
getPathInfo() : String
getQueryString() : String
getRemoteUser() : String
getUserPrincipal() : java.security.Principal
isUserInRole(role : String) : boolean
getRequestedSessionId() : String
getRequestURI() : String
getRequestURL() : String
getSession() : HttpSession
getSession(create : boolean) : HttpSession
<<interface>>
java.security.Principal
getName(): String
equals(Object another): boolean
hashCode() : int
toString() : String

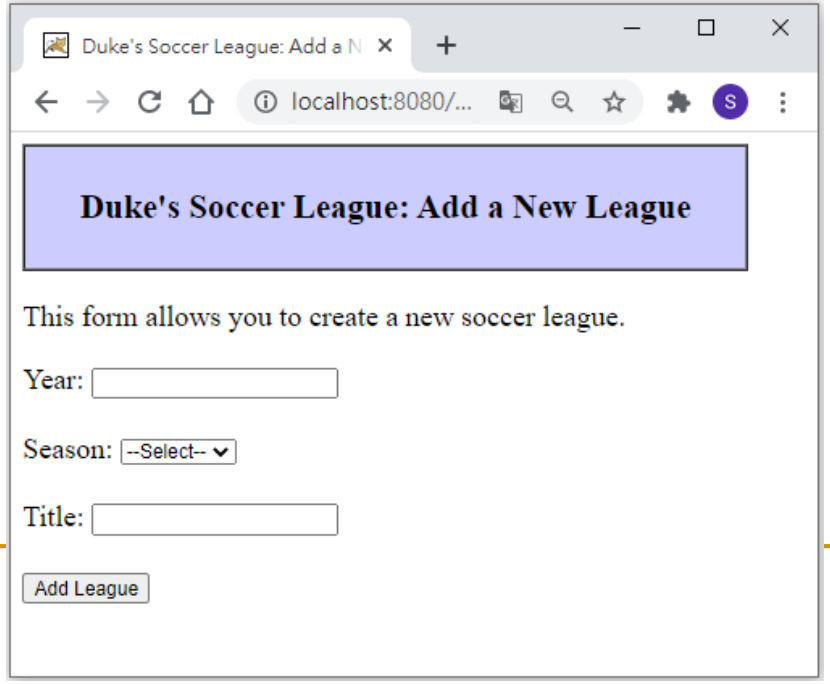
HttpServletRequest – 使用者身份

方法名稱	回傳型態	用途說明
getRemoteUser()	String	取得 Login 名稱
getUserPrincipal()	java.security.Principal	取得目前授權使用者的 Principal 物件(代表個人、群組或登錄 id 主體身份), 若為驗證通過,傳回null值
isUserInRole(role : String)	boolean	判斷目前授權使用者是否屬於指定角色(群組)


Duke's Soccer League: Home
localhost:8080/Socc...
Duke's Soccer League: Home
This is the Home page for Duke's Soccer League.
Players
List all leagues
Register for a league
League Administrator
Add a new league


Duke's Soccer League: Login
localhost:8080/...
Duke's Soccer League: Login
Please login to the web application.
User Name: Sean
Password:
Login

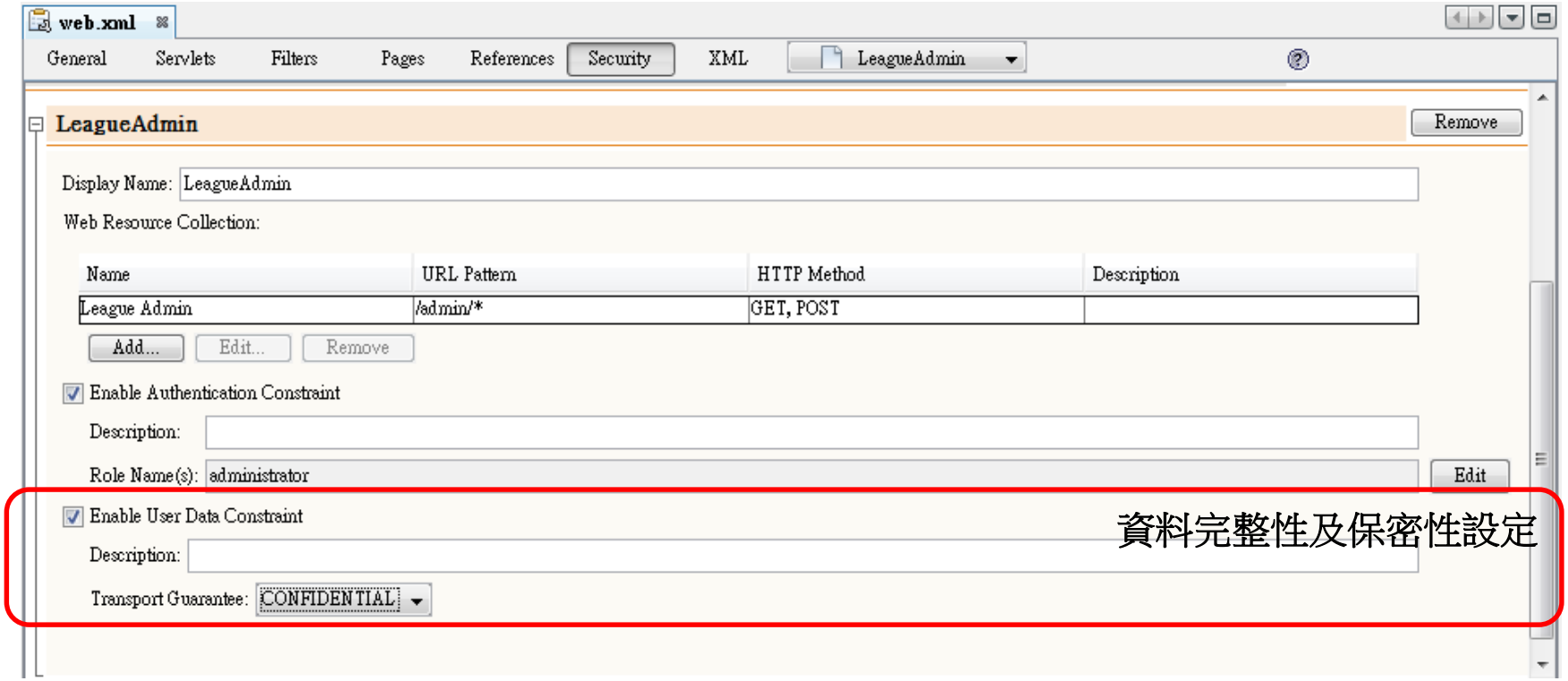

Duke's Soccer League: Login Fa
localhost:8080/Socc...
Duke's Soccer League: Login Failure
Sorry, your login request has failed.


Duke's Soccer League: Add a N
localhost:8080/...
Duke's Soccer League: Add a New League
This form allows you to create a new soccer league.
Year:
Season: --Select--
Title:
Add League

課程大綱

1) 網路安全機制
2) 網路資源授權機制
3) 身份驗證機制
4) **資料完整性及保密性機制**

標籤名稱	出現次數	說明
<security-constraint>	*	安全性限制
<display-name>	?	安全資源名稱
<web-resource-collection>	+	受保護的網路資源集合
<web-resource-name>	1	網路資源集合名稱
<description>	?	資源描述
<url-pattern>	*	資源集合所在的URL
<http-method>	*	受保護的HTTP方法 通過授權者才可使用該方法存取該資源 未設定此標籤則所有連線方法皆不可使用
<auth-constraint >	?	角色限制設定
<description>	?	限制描述
<role-name>	*	可存取資源的安全性角色 對應<security-role>標籤所定義的角色
<user-data-constraint>	?	傳輸限制設定
<description>	?	傳輸限制描述
<transport-guarantee>	1	通訊層設定的保護方式 NONE:不保護保密性及完整性 INTEGRAL:確保資料完整性 CONFIDENTIAL:確保資料保密性


web.xml
General
Servlets
Filters
Pages
References
Security
XML
LeagueAdmin
LeagueAdmin
Remove
Display Name: LeagueAdmin
Web Resource Collection:
Name
URL Pattern
HTTP Method
Description
League Admin
/admin/*
GET, POST
Add...
Edit...
Remove
Enable Authentication Constraint
Description:
Role Name(s): administrator
Edit
Enable User Data Constraint
Description:
Transport Guarantee: CONFIDENTIAL
資料完整性及保密性設定

Lab 2

- 修改登入機制，使用自訂登入畫面
  - 新增 login.jsp
    - Form Action = “j_security_check”
    - User Name 欄位名稱為 j_username
    - Password欄位,名稱為j_password
    - 提供 submit 及 reset 按鈕, 分別顯示為 login 及 clear
  - 編輯 web.xml, 使用 login.jsp 作為登入頁面
    - 展開Login Configuration, 選擇 form
    - Form Login Page 選擇 login.jsp
    - Form Error Page 也選擇 login.jsp

Lab 2

- 取得使用者登入名稱
  - 修改 web.Controller.java
    - 取得session 物件
    - 若session中無user屬性
      - 由request請求中取得UserPrincipal物件,
      - 由Principal物件取得name
      - 加入session屬性(user, name)
  - 修改 customerList.jsp 及 customerView.jsp 檔案
    - 在內容之前,顯示問候user資訊
- 測試、執行