
練習

網路應用程式裡使用Filter

鄭安翔

ansel_cheng@hotmail.com

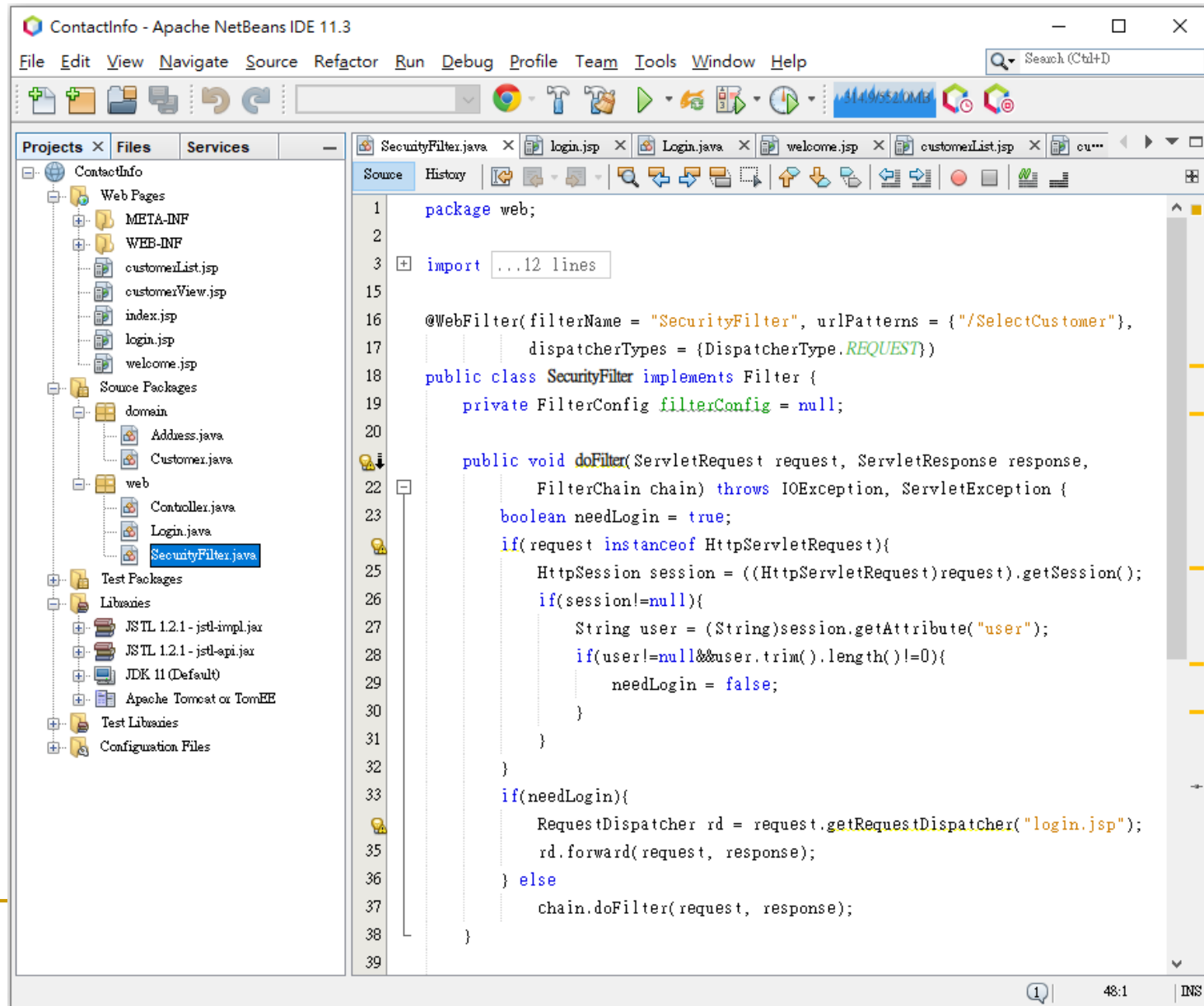
Lab

- 修改 **ContactInfo** 網路應用專案
- 新增 **Filter** 類別 **SecurityFilter.java**
 - Filter Mapping URL Patterns 設為 `/SelectCustomer`
 - Dispatch 狀態勾選 **REQUEST**
 - 初始參數：無
 - 刪除下列 **NetBeans** 實作內容
 - `getFilterConfig()`、`setFilterConfig()`、`sendProcessError()`、`log()`、`toString()`、`doBeforeProcessing()`、`doAfterProcessing()` 方法
 - `init(FilterConfig filterConfig)` 方法僅保留 `filterConfig` 初始設定
 - `destroy()` 方法修改為空白實作
 - 修改 `doFilter(...)`
 - 取得 **Session** 物件中 `"user"` 屬性
 - 若 **Session** 物件或 `"user"` 屬性不存在時,轉送 `login.jsp` 畫面
 - **Session** 物件及 `"user"` 屬性存時,執行 `chain.doFilter(...)`;

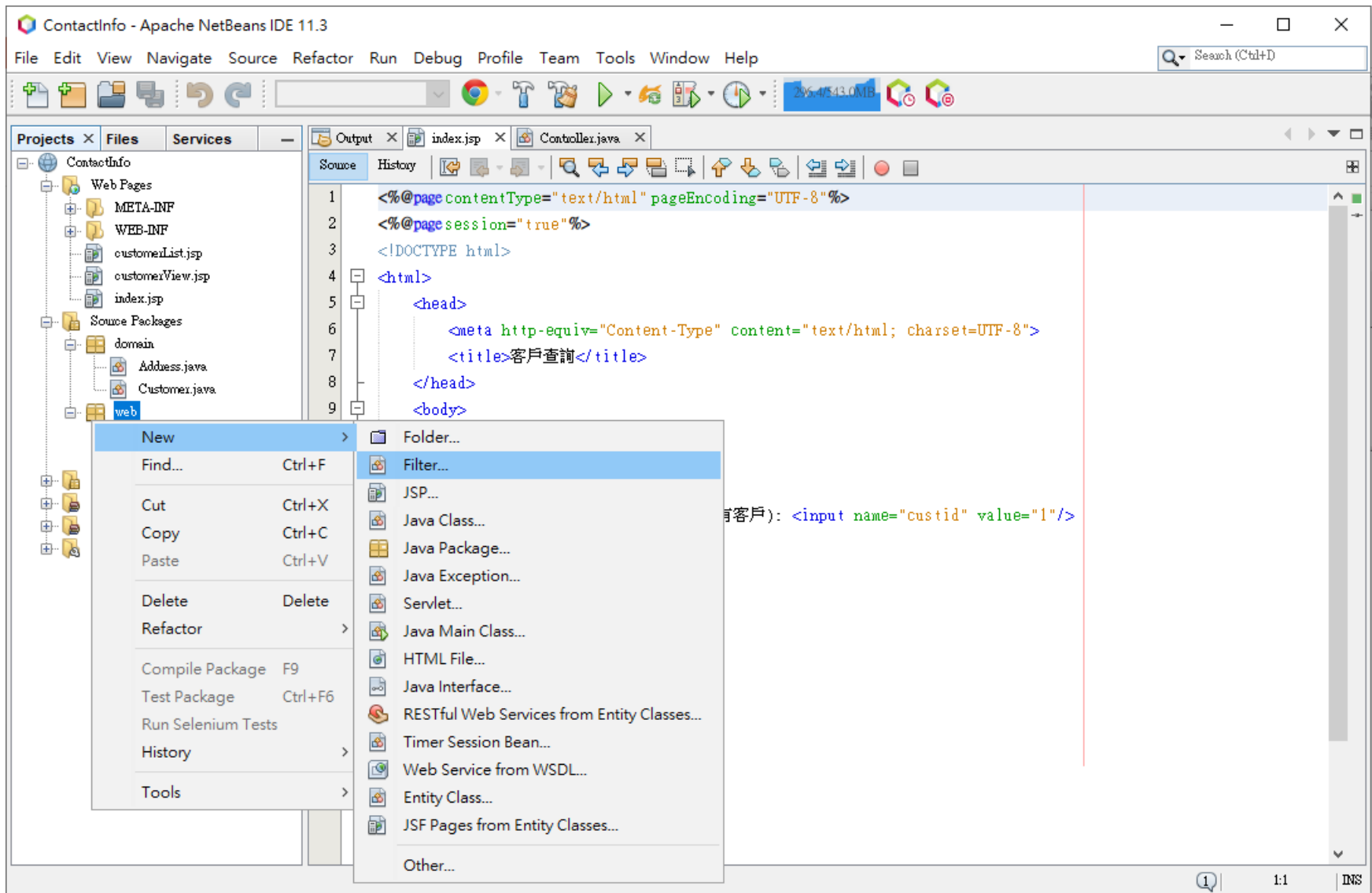
Lab

- 新增 login.jsp
 - 輸入使用者姓名的Form表單
 - Action 送至"Login"
 - user 輸入欄位
- 新增 web.Login.java Servlet
 - 取得請求參數user並存入HttpSession屬性
 - 轉送至welcome.jsp
- 新增 welcome.jsp
 - 顯示歡迎使用者登入資訊
 - 回首頁連結
- 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
 - 顯示問候使用者的資訊

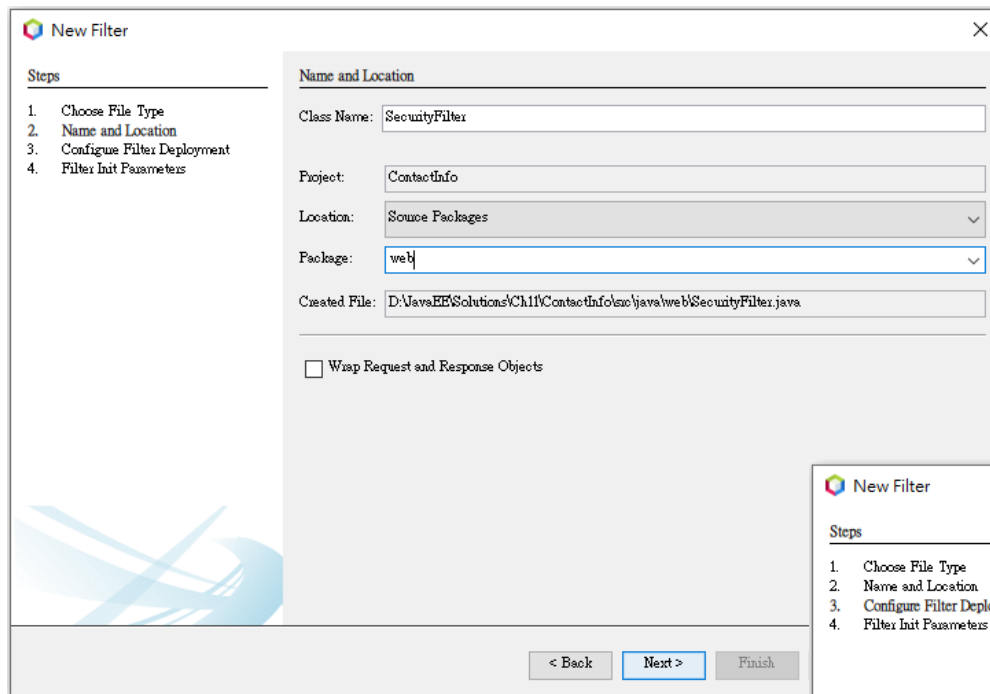
ContactInfo 專案



新增Filter 類別 SecurityFilter.java



新增Filter 類別 SecurityFilter.java



New Filter

Steps

1. Choose File Type
2. Name and Location
3. Configure Filter Deployment
4. Filter Init Parameters

Name and Location

Class Name:

Project:

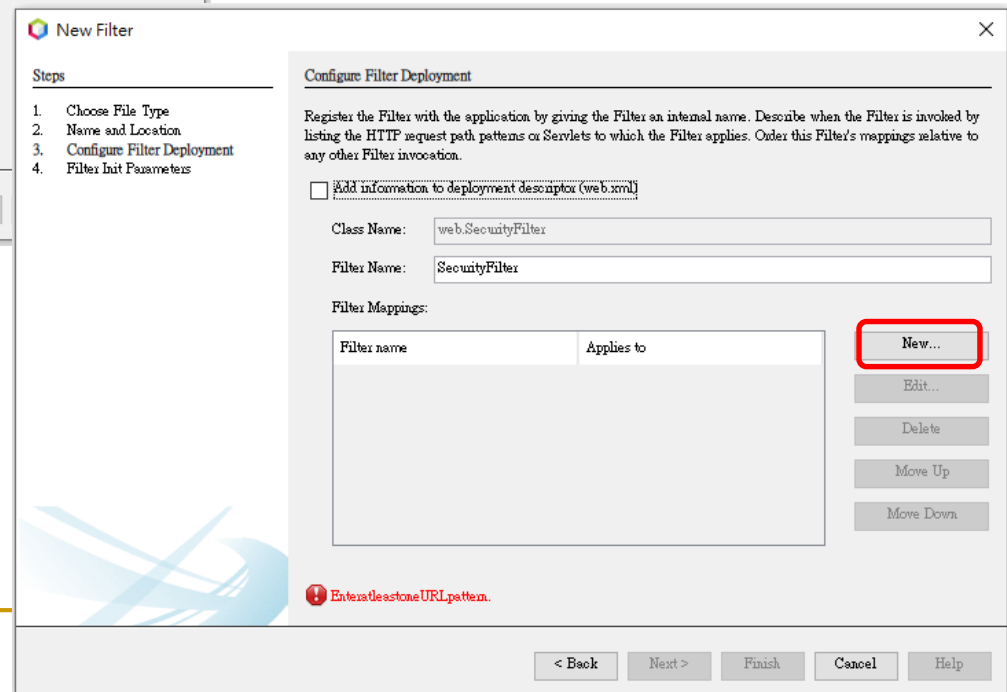
Location:

Package:

Created File:

☐ Wrap Request and Response Objects

< Back Next > Finish



New Filter

Steps

1. Choose File Type
2. Name and Location
3. Configure Filter Deployment
4. Filter Init Parameters

Configure Filter Deployment

Register the Filter with the application by giving the Filter an internal name. Describe when the Filter is invoked by listing the HTTP request path patterns or Servlets to which the Filter applies. Order this Filter's mappings relative to any other Filter invocation.

☐ Add information to deployment descriptor (web.xml)

Class Name:

Filter Name:

Filter Mappings:

Filter name	Applies to
-------------	------------

New...
Edit...
Delete
Move Up
Move Down

Enter at least one URL pattern.

< Back Next > Finish Cancel Help

新增Filter 類別 SecurityFilter.java

Filter Mapping

Filter Name:

☒ URL:

☐ Servlet:

Dispatch Conditions

☒ REQUEST ☐ FORWARD ☐ INCLUDE ☐ ERROR

OK Cancel

New Filter

Steps

1. Choose File Type
2. Name and Location
3. Configure Filter Deployment
4. Filter Init Parameters

Configure Filter Deployment

Register the Filter with the application by giving the Filter an internal name. Describe when the Filter is invoked by listing the HTTP request path patterns or Servlets to which the Filter applies. Order this Filter's mappings relative to any other Filter invocation.

☐ Add information to deployment descriptor (web.xml)

Class Name:

Filter Name:

Filter Mappings:

Filter name	Applies to
SecurityFilter	/SelectCustomer

New... Edit... Delete Move Up Move Down

< Back Next > Finish Cancel Help

New Filter

Steps

1. Choose File Type
2. Name and Location
3. Configure Filter Deployment
4. Filter Init Parameters

Filter Init Parameters

Specify any init parameters for the Filter.

Initialization Parameters:

Name	Value
------	-------

New Edit... Delete

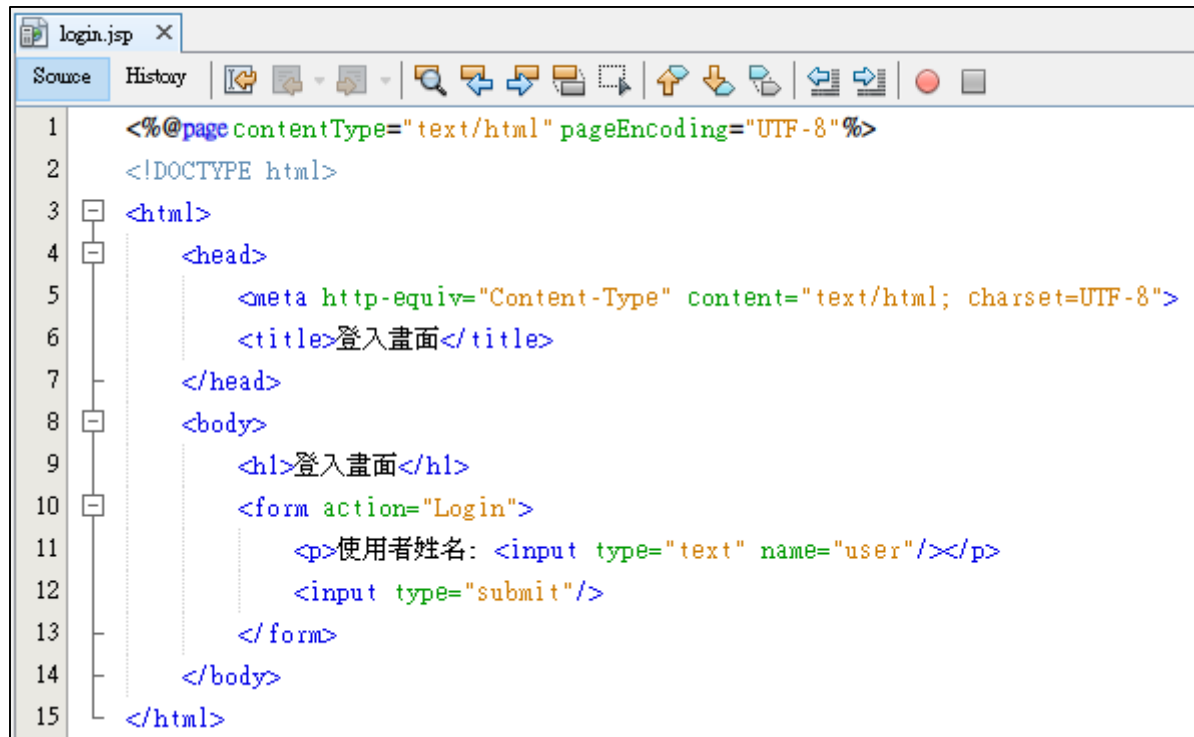
< Back Next > Finish Cancel Help

SecurityFilter.java

```
SecurityFilter.java x
Source History
1 package web;
2
3 import ...12 lines
15
16 @WebFilter(filterName = "SecurityFilter", urlPatterns = {"/SelectCustomer"},
17           dispatcherTypes = {DispatcherType.REQUEST})
18 public class SecurityFilter implements Filter {
19     private FilterConfig filterConfig = null;
20
21     public void doFilter(ServletRequest request, ServletResponse response,
22                         FilterChain chain) throws IOException, ServletException {
23         boolean needLogin = true;
24         if(request instanceof HttpServletRequest){
25             HttpSession session = ((HttpServletRequest)request).getSession();
26             if(session!=null){
27                 String user = (String)session.getAttribute("user");
28                 if(user!=null&&user.trim().length()!=0){
29                     needLogin = false;
30                 }
31             }
32         }
33         if(needLogin){
34             RequestDispatcher rd = request.getRequestDispatcher("login.jsp");
35             rd.forward(request, response);
36         } else
37             chain.doFilter(request, response);
38     }
}
```

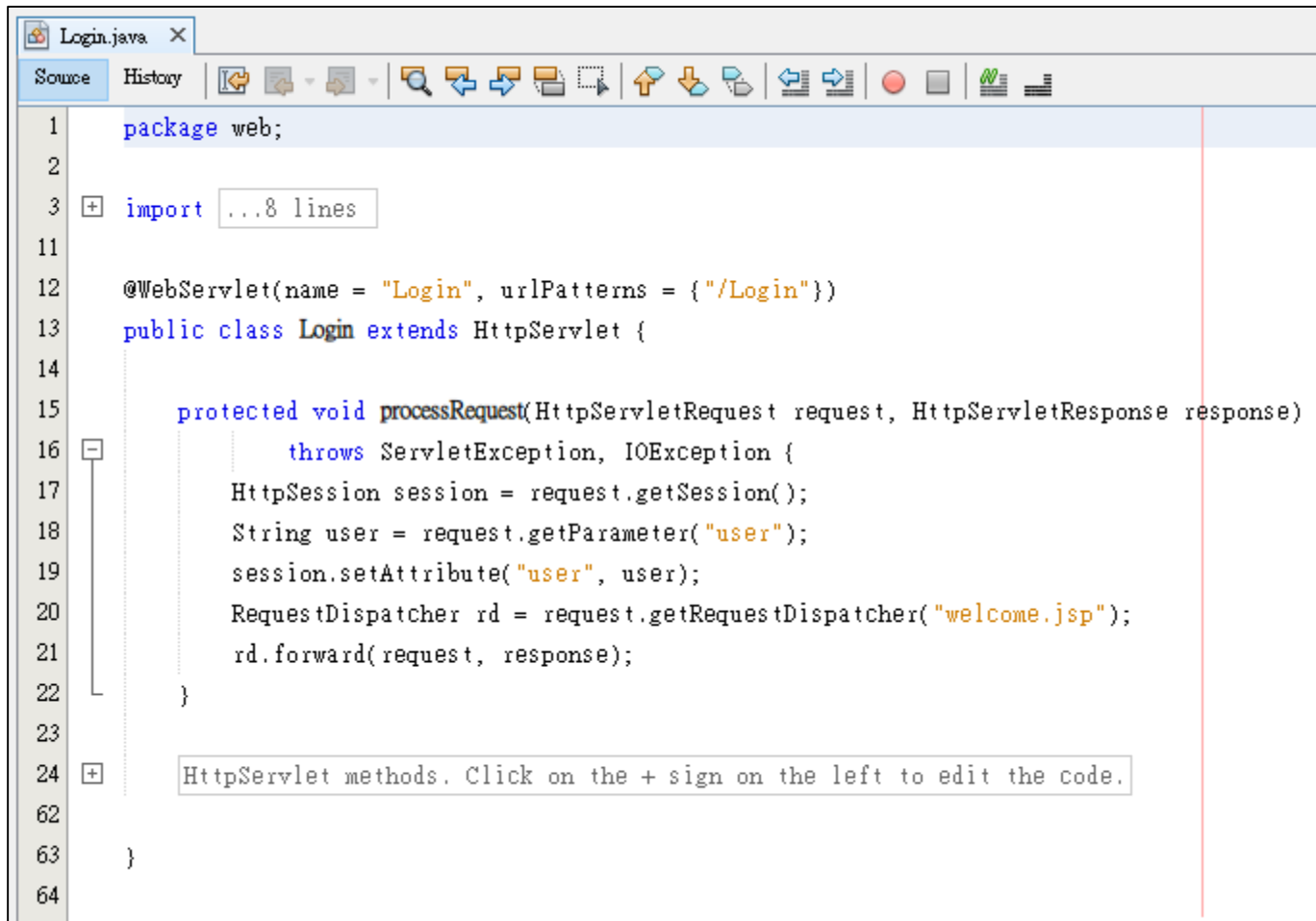
```
39
40
41     public void destroy() {
42     }
43
44     public void init(FilterConfig filterConfig) {
45         this.filterConfig = filterConfig;
46     }
47 }
```


login.jsp



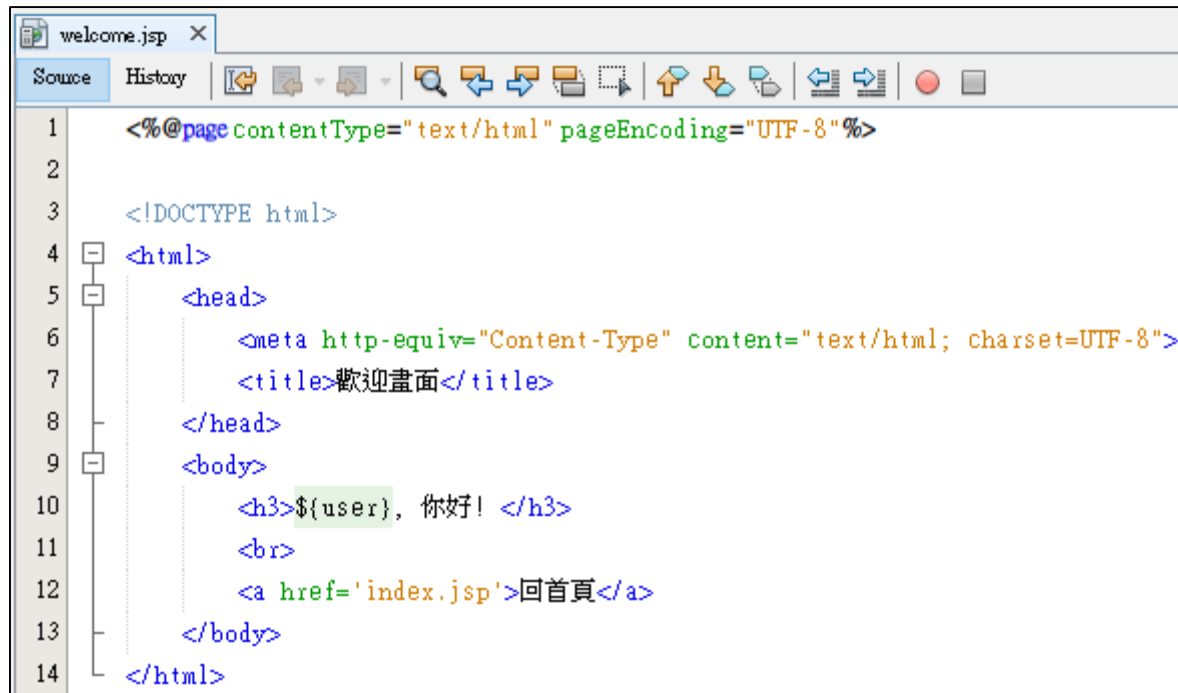
```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4 <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <title>登入畫面</title>
7 </head>
8 <body>
9     <h1>登入畫面</h1>
10    <form action="Login">
11        <p>使用者姓名: <input type="text" name="user"/></p>
12        <input type="submit"/>
13    </form>
14 </body>
15 </html>
```

web.Login.java



```
1 package web;
2
3 import ...8 lines
11
12 @WebServlet(name = "Login", urlPatterns = {"/Login"})
13 public class Login extends HttpServlet {
14
15     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
16         throws ServletException, IOException {
17         HttpSession session = request.getSession();
18         String user = request.getParameter("user");
19         session.setAttribute("user", user);
20         RequestDispatcher rd = request.getRequestDispatcher("welcome.jsp");
21         rd.forward(request, response);
22     }
23
24     HttpServlet methods. Click on the + sign on the left to edit the code.
62
63 }
64
```

welcome.jsp



```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7     <title>歡迎畫面</title>
8   </head>
9   <body>
10    <h3>${user}, 你好! </h3>
11    <br>
12    <a href='index.jsp'>回首頁</a>
13  </body>
14 </html>
```

customerView.jsp

```
customerView.jsp x
Source History
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3
4 <!DOCTYPE html>
5 <html>
6 <head>
7     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8     <title>Customer Info</title>
9 </head>
10 <body>
11     <h3>${user}, 你好!</h3>
12     <h3>客戶資訊</h3>
```

```
13
14
15
16
17
18
19
20
21
22
23
24
<c:choose>
    <c:when test = "${not empty customer}">
        <p>姓名: ${customer.name}</p>
        <cif test = "${not empty customer.officeAddress}">
            <p>辦公室地址:<br>
                ${customer.officeAddress.address1},<br>
                ${customer.officeAddress.address2},<br>
                ${customer.officeAddress.city},<br>
                ${customer.officeAddress.country},<br>
                ${customer.officeAddress.postcode}<br>
            </p>
        </cif>
        <cif test = "${customer.billingAddress ne null}">
            <p>發票地址:<br>
                ${customer["billingAddress"].address1},<br>
                ${customer["billingAddress"].address2},<br>
                ${customer["billingAddress"].city},<br>
                ${customer["billingAddress"].country},<br>
                ${customer["billingAddress"].postcode}<br>
            </p>
        </cif>
        <cif test = "${not empty customer.addresses[2]}">
            <p>送貨地址:<br>
                ${customer.addresses[2].address1},<br>
                ${customer.addresses[2].address2},<br>
                ${customer.addresses[2].city},<br>
                ${customer["addresses"][2].country},<br>
                ${customer["addresses"][2].postcode}<br>
            </p>
        </cif>
    </c:when>
    <c:otherwise>
        <h3>查詢的客戶不存在</h3>
    </c:otherwise>
</c:choose>
</body>
</html>
```

customerList.jsp

```
customerList.jsp x
Source History
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7 <title>Customer List</title>
8 </head>
9 <body>
10 <h3>${user}, 你好!</h3>
11 <h3>客戶列表</h3>
```

```
12 <c:choose>
13 <c:when test = "${not empty customers}">
14 <c:forEach var="customer" items="${customers}">
15 <p>姓名: ${customer.name}</p>
16 <cif test = "${not empty customer.officeAddress}">
17 <p>辦公室地址:
18     ${customer.officeAddress.address1},
19     ${customer.officeAddress.address2},
20     ${customer.officeAddress.city},
21     ${customer.officeAddress.country},
22     ${customer.officeAddress.postcode}
23 </p>
24 </c:forEach>
25 <cif test = "${customer.billingAddress ne null}">
26 <p>發票地址:
27     ${customer["billingAddress"].address1},
28     ${customer["billingAddress"].address2},
29     ${customer["billingAddress"].city},
30     ${customer["billingAddress"].country},
31     ${customer["billingAddress"].postcode}
32 </p>
33 </c:forEach>
34 <cif test = "${not empty customer.addresses[2]}">
35 <p>送貨地址:
36     ${customer.addresses[2].address1},
37     ${customer.addresses[2].address2},
38     ${customer.addresses[2].city},
39     ${customer["addresses"][2].country},
40     ${customer["addresses"][2].postcode}
41 </p>
42 </c:forEach>
43 </c:when>
44 <c:otherwise>
45 <h3>無客戶存在</h3>
46 </c:otherwise>
47 </c:choose>
48 </body>
49 </html>
```

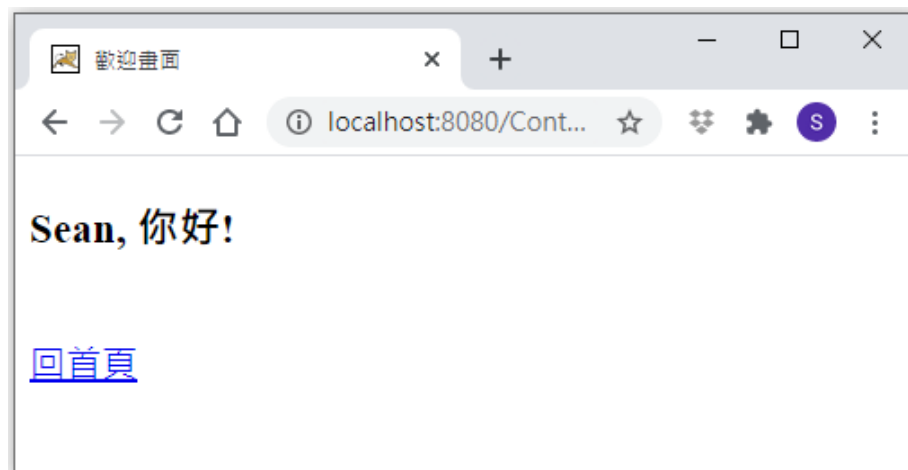
測試、執行



客戶查詢

輸入客戶編號(輸入0查詢所有客戶):

提交



歡迎畫面

Sean, 你好!

[回首頁](#)



登入畫面

使用者姓名:

提交

測試、執行

