
Java EE 7 Web 練習四 Solution

鄭安翔

ansel_cheng@hotmail.com

Lab 1

- 使用NetBeans IDE，建立 Anniversary 網路應用專案

- 設計Model類別AnniversaryModel.java

- 結婚周年與材質名稱對照表如右表
- 定義適當的屬性及方法
- 符合JavaBeans規則

- 建立index.html

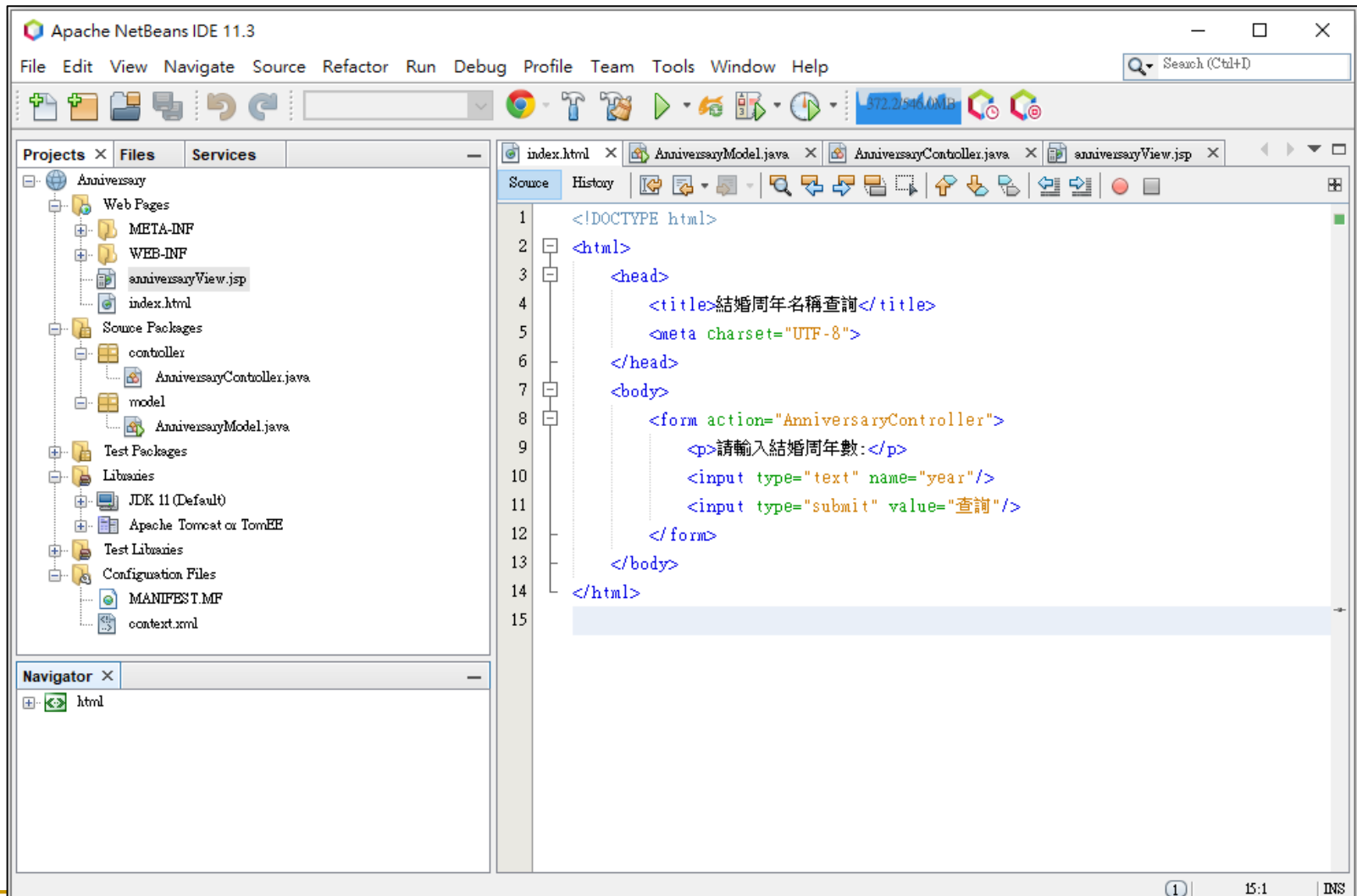
- 使用表單Form, 供使用者輸入結婚周年數
- 將表單內容傳送到AnniversaryController (Servlet)

周年	材質名稱
1	Paper
2	Cotton
3	Leather
4	Linen, Silk
5	Wood
6	Iron
7	Wool, Copper
8	Bronze
9	Pottery, China
10	Tin, Aluminum
11	Steel
12	Silk
13	Lace
14	Ivory
15	Crystal
20	China
25	Silver
30	Pearl
35	Coral
40	Ruby
45	Sapphire
50	Gold
55	Emerald
60	Diamond

Lab 1

- 建立AnniversaryController.java
 - 建立AnniversaryModel物件
 - 取的結婚周年數請求參數,設定Model物件year 欄位
 - 將Model物件設為請求Attribute
 - 轉送至anniversaryView.jsp
- 建立 anniversaryView.jsp
 - 使用EL, 取得AnniversaryModel的year及material屬性值
 - 顯示於網頁
- 測試、執行

建立 Anniversary 專案

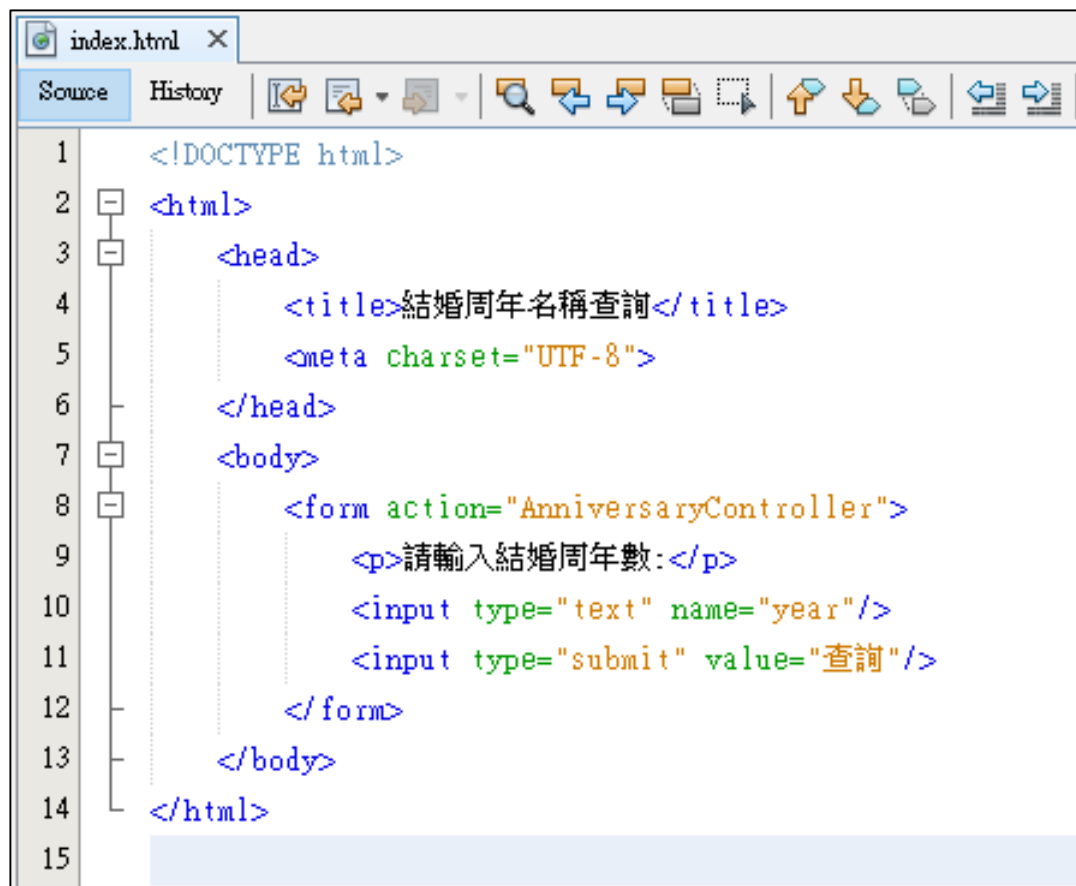


建立AnniversaryModel.java

```
AnniversaryModel.java x
Source History
1 package model;
2
3 import java.util.*;
4
5 public class AnniversaryModel {
6
7     private int year;
8     private Map<Integer, String> anniversaryMap;
9
10    public AnniversaryModel() {
11        anniversaryMap = new HashMap<Integer, String>();
12        populateMap();
13    }
14
15    public int getYear() {
16        return year;
17    }
18
19    public void setYear(int year) {
20        this.year = year;
21    }
22
23    public String getMaterial(){
24        String result = "Nothing";
25        if(anniversaryMap.containsKey(year)){
26            result = anniversaryMap.get(year);
27        }
28        return result;
29    }
30}
```

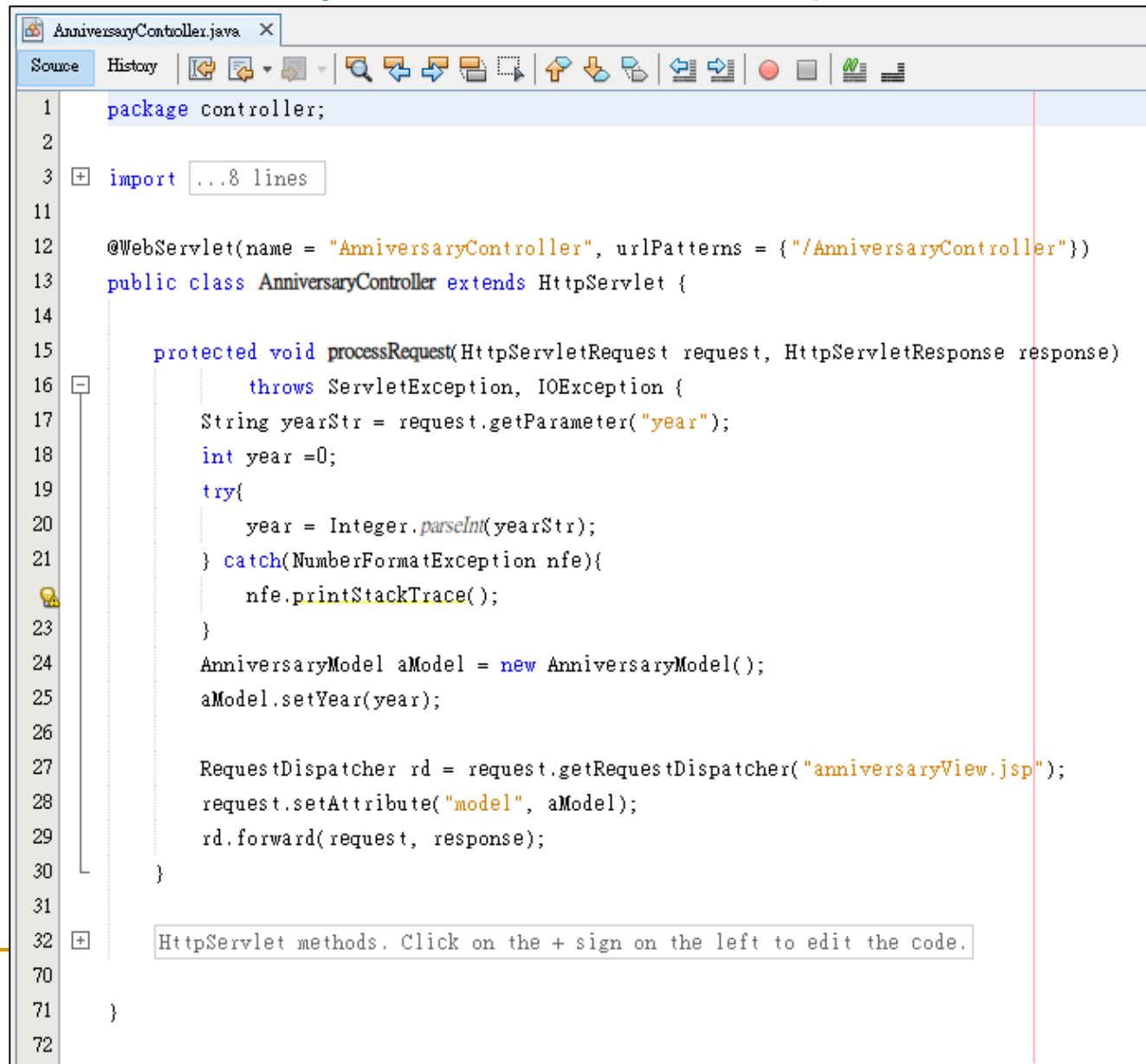
```
31 private void populateMap(){
32     anniversaryMap.put(1, "Paper");
33     anniversaryMap.put(2, "Cotton");
34     anniversaryMap.put(3, "Leather");
35     anniversaryMap.put(4, "Linen, Silk");
36     anniversaryMap.put(5, "Wood");
37     anniversaryMap.put(6, "Iron");
38     anniversaryMap.put(7, "Wool, Copper");
39     anniversaryMap.put(8, "Bronze");
40     anniversaryMap.put(9, "Pottery, China");
41     anniversaryMap.put(10, "Tin, Aluminum");
42     anniversaryMap.put(11, "Steel");
43     anniversaryMap.put(12, "Silk");
44     anniversaryMap.put(13, "Lace");
45     anniversaryMap.put(14, "Ivory");
46     anniversaryMap.put(15, "Crystal");
47     anniversaryMap.put(20, "China");
48     anniversaryMap.put(25, "Silver");
49     anniversaryMap.put(30, "Pearl");
50     anniversaryMap.put(35, "Coral");
51     anniversaryMap.put(40, "Ruby");
52     anniversaryMap.put(45, "Sapphire");
53     anniversaryMap.put(50, "Gold");
54     anniversaryMap.put(55, "Emerald");
55     anniversaryMap.put(60, "Diamond");
56 }
57 }
```

index.html

A screenshot of a web browser's source code view for a file named 'index.html'. The browser window has a title bar with the file name and a close button. Below the title bar is a toolbar with various icons for navigating and editing the source code. The source code is displayed in a text area with line numbers on the left. The code is an HTML document with a DOCTYPE declaration, a head section containing a title and a meta charset declaration, and a body section containing a form with a text input and a submit button. The form's action is set to 'AnniversaryController'. The title is in Chinese, and the submit button value is also in Chinese.

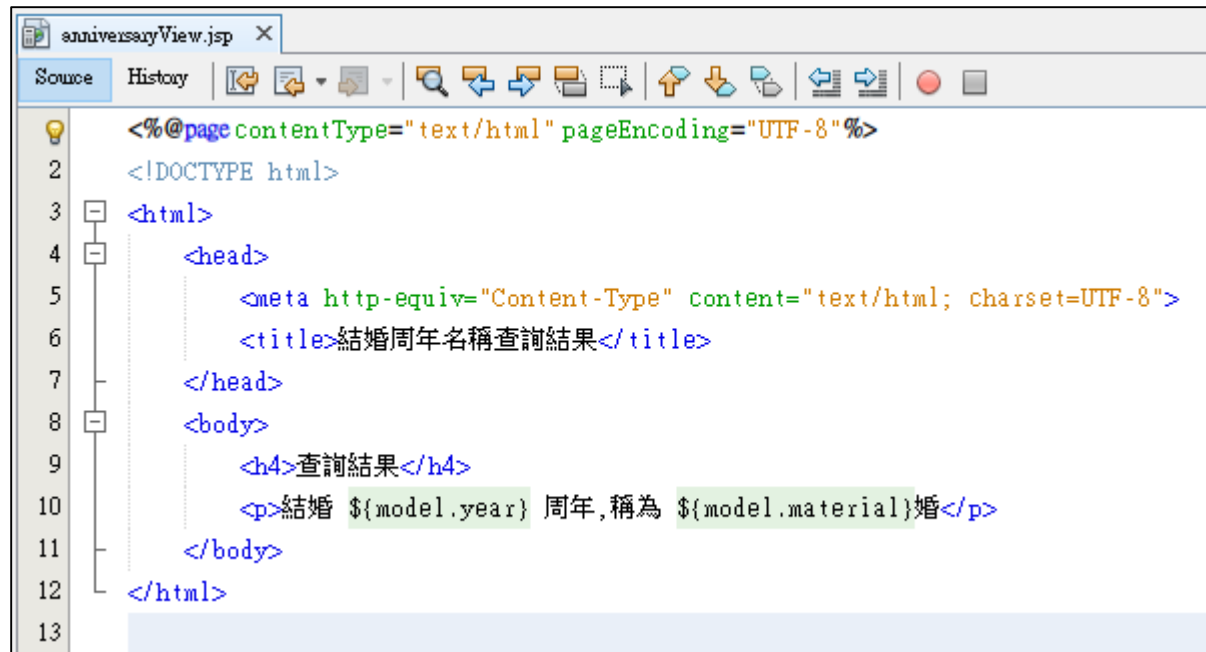
```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>結婚周年名稱查詢</title>
5     <meta charset="UTF-8">
6   </head>
7   <body>
8     <form action="AnniversaryController">
9       <p>請輸入結婚周年數:</p>
10      <input type="text" name="year"/>
11      <input type="submit" value="查詢"/>
12    </form>
13  </body>
14 </html>
15
```

AnniversaryController.java



```
1 package controller;
2
3 import ...8 lines
11
12 @WebServlet(name = "AnniversaryController", urlPatterns = {"/AnniversaryController"})
13 public class AnniversaryController extends HttpServlet {
14
15     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
16         throws ServletException, IOException {
17         String yearStr = request.getParameter("year");
18         int year = 0;
19         try{
20             year = Integer.parseInt(yearStr);
21         } catch(NumberFormatException nfe){
22             nfe.printStackTrace();
23         }
24         AnniversaryModel aModel = new AnniversaryModel();
25         aModel.setYear(year);
26
27         RequestDispatcher rd = request.getRequestDispatcher("anniversaryView.jsp");
28         request.setAttribute("model", aModel);
29         rd.forward(request, response);
30     }
31
32     HttpServlet methods. Click on the + sign on the left to edit the code.
70
71 }
72
```

anniversaryView.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>結婚周年名稱查詢結果</title>
7  </head>
8  <body>
9      <h4>查詢結果</h4>
10     <p>結婚 ${model.year} 周年,稱為 ${model.material}婚</p>
11 </body>
12 </html>
13
```


測試、執行



結婚周年名稱查詢

localhost:8080/Anniversary/

請輸入結婚周年數:

10 查詢



結婚周年名稱查詢結果

localhost:8080/Anniversar...

查詢結果

結婚 10 周年,稱為 Tin, Aluminum婚

Lab2 DVDLibrary 分析設計

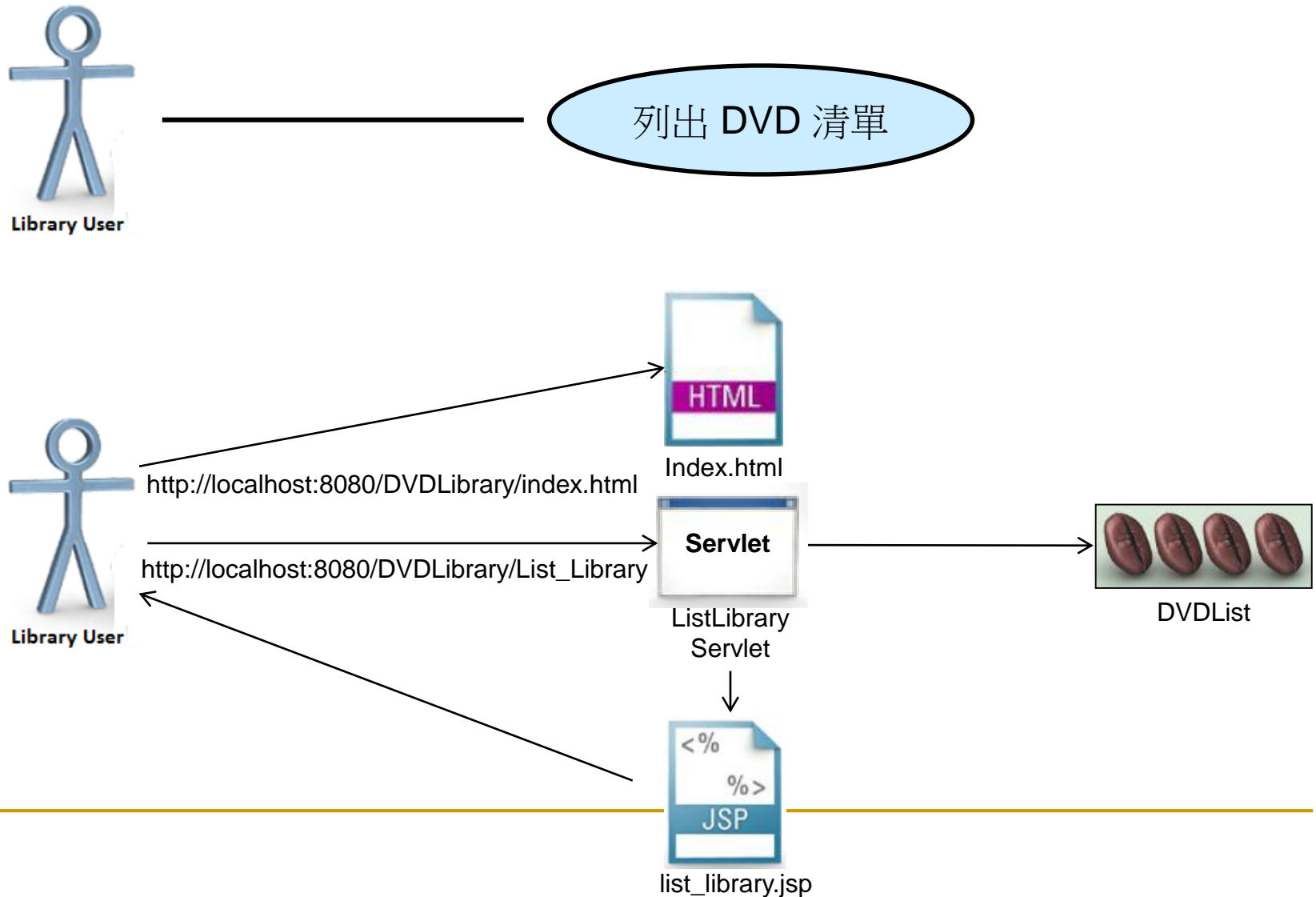


DVD Library 網路應用程式

列出 DVD 清單

新增 DVD 加入清單

Lab2 DVDLibrary 應用程式



Lab2

- 使用NetBeans IDE，建立DVDLibrary 網路應用專案
- 編輯 index.html
 - 提供一個連結,連結至List_Library
 - 畫面如下



Lab2

■ 撰寫model.DVDItem.java

- 模型物件設計如右：
- 定義三個String屬性：title, year, genre
- 提供建構子及屬性 getter

DVDItem
-title : String -year : String -genre : String
«constructor» DVDItem(title:String, year:String, genre:String)
«accessors» getYear() : String getTitle() : String getGenre() : String

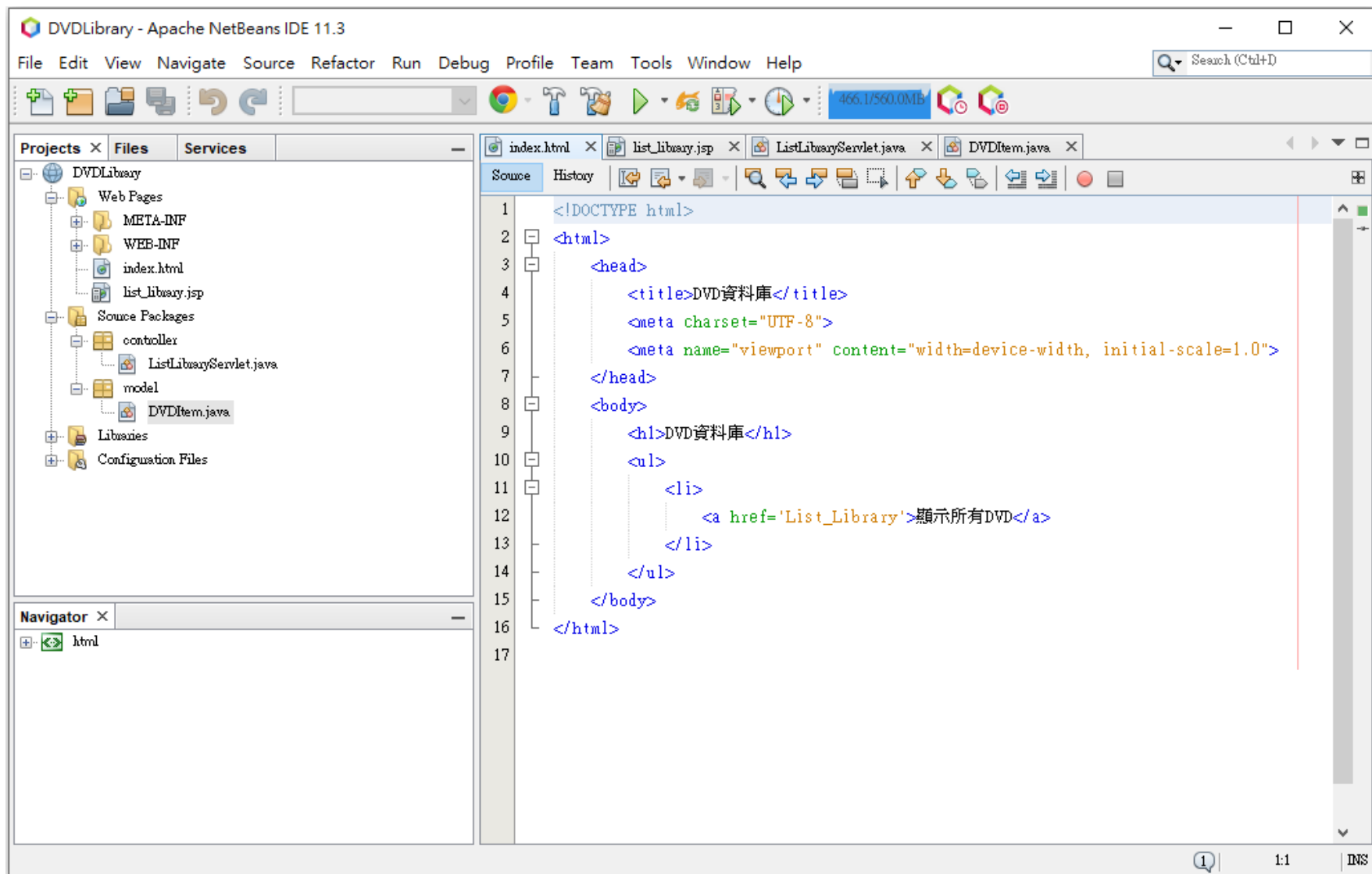
■ 撰寫controller.ListLibraryServlet.java

- Servlet URL 為 List_Library
 - 撰寫 private List<DVDItem> createDvdList() 方法
 - 建立三個DVDItem物件
 - 加入List清單後傳回
 - processRequest()方法中呼叫createDvdList(),取得DVD清單
 - 將DVD清單加入請求(Request)中DVDList屬性(attribute)
-
- 請求轉送至list_library.jsp

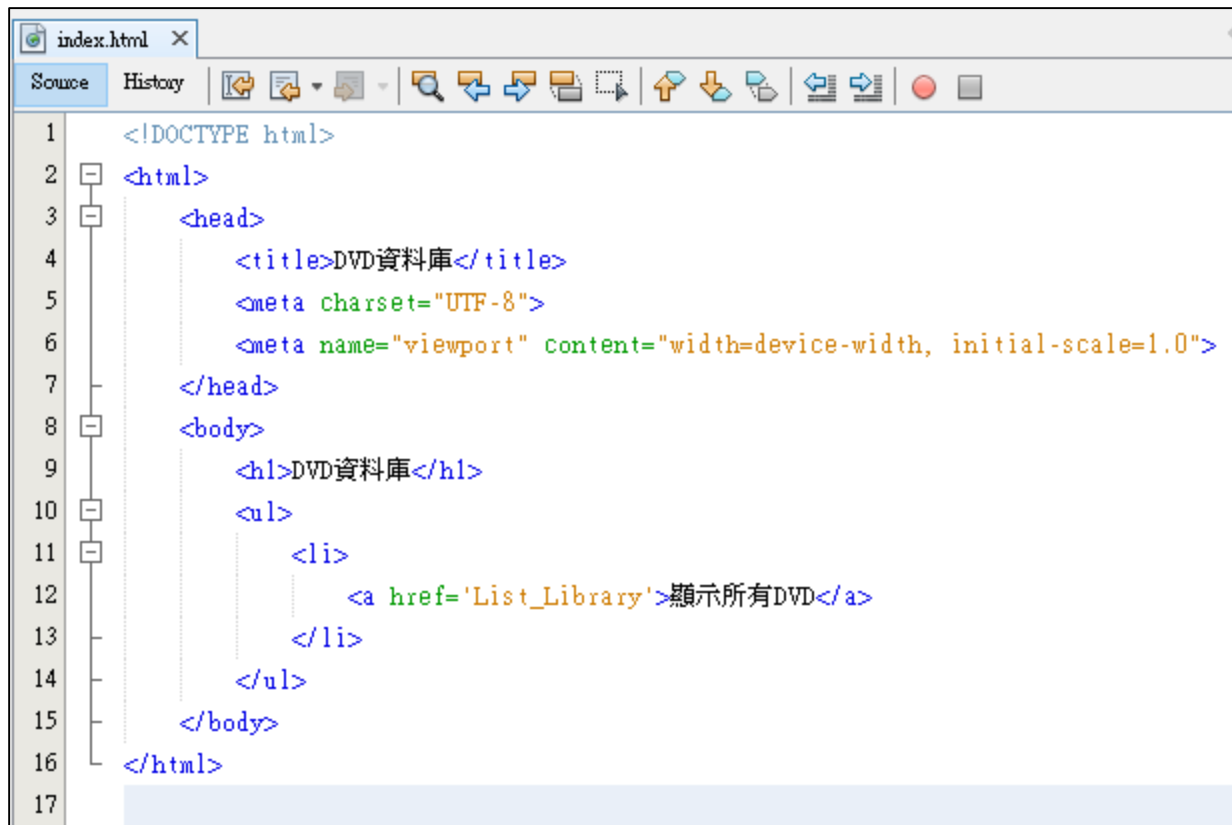
Lab2

- 撰寫list_library.jsp
 - 使用EL取得DVDList屬性值顯示於網頁
- 測試、執行

建立DVDLibrary專案



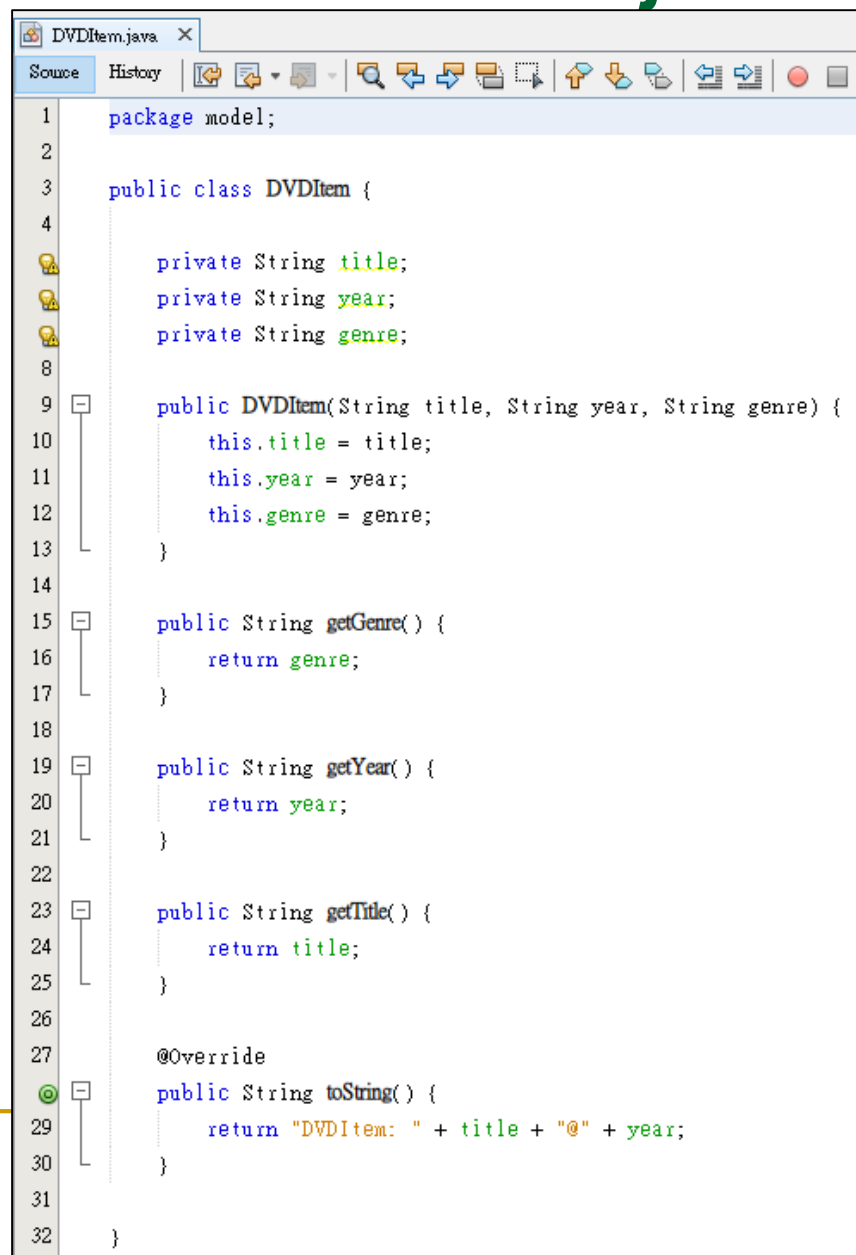
index.html



The image shows a web browser window with the source code of a file named 'index.html'. The browser's address bar is empty, and the page title is 'index.html'. The source code is displayed in a monospaced font with syntax highlighting. The code is as follows:

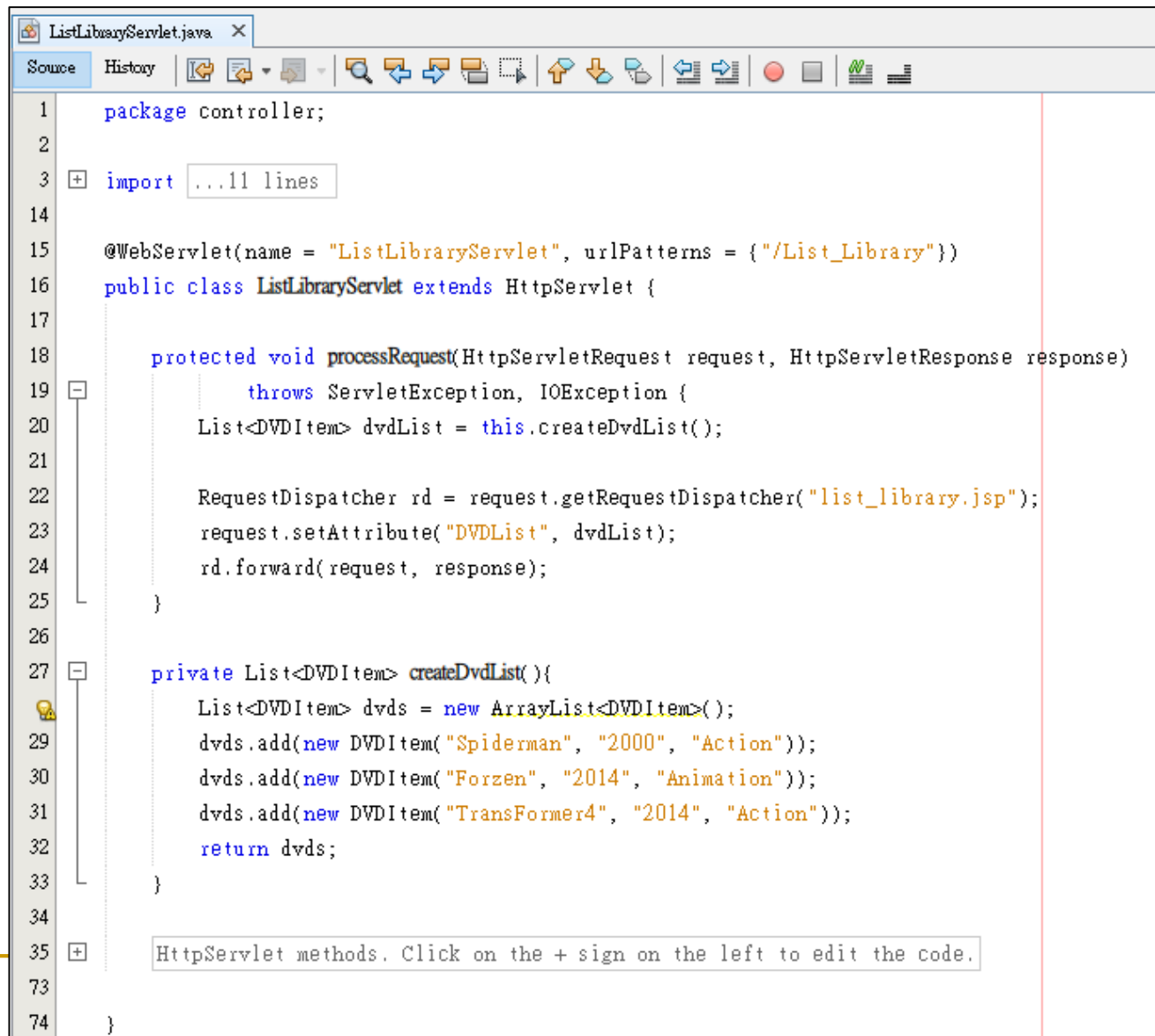
```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>DVD資料庫</title>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  </head>
8  <body>
9      <h1>DVD資料庫</h1>
10     <ul>
11         <li>
12             <a href='List_Library'>顯示所有DVD</a>
13         </li>
14     </ul>
15 </body>
16 </html>
17
```


建立model.DVDItem.java

A screenshot of an IDE window titled 'DVDItem.java'. The window has a 'Source' tab and a toolbar with various icons. The code is written in Java and defines a 'DVDItem' class with private attributes 'title', 'year', and 'genre'. It includes a constructor and four getter methods. The '@Override' annotation is present above the 'toString' method. The code is as follows:

```
1 package model;
2
3 public class DVDItem {
4
5     private String title;
6     private String year;
7     private String genre;
8
9     public DVDItem(String title, String year, String genre) {
10         this.title = title;
11         this.year = year;
12         this.genre = genre;
13     }
14
15     public String getGenre() {
16         return genre;
17     }
18
19     public String getYear() {
20         return year;
21     }
22
23     public String getTitle() {
24         return title;
25     }
26
27     @Override
28     public String toString() {
29         return "DVDItem: " + title + "@" + year;
30     }
31
32 }
```

controller.ListLibraryServlet.java



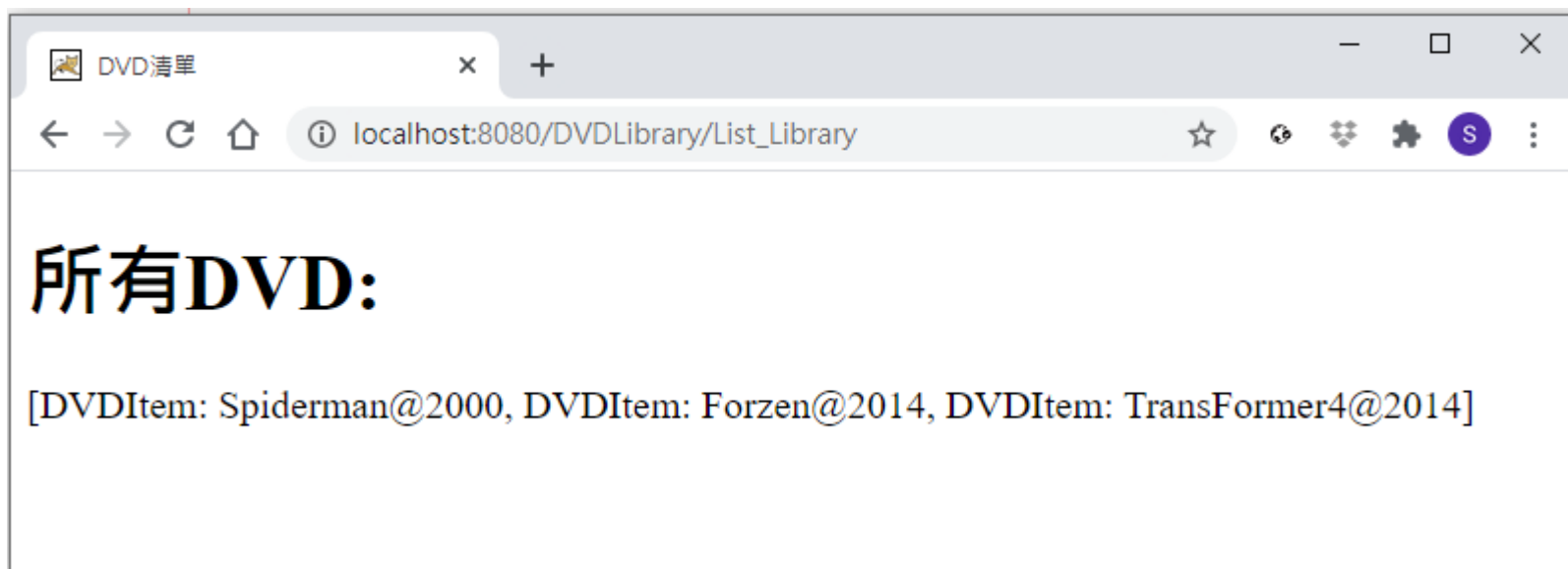
```
1 package controller;
2
3 import ...11 lines
4
14
15 @WebServlet(name = "ListLibraryServlet", urlPatterns = {"/List_Library"})
16 public class ListLibraryServlet extends HttpServlet {
17
18     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
19         throws ServletException, IOException {
20         List<DVDItem> dvdList = this.createDvdList();
21
22         RequestDispatcher rd = request.getRequestDispatcher("list_library.jsp");
23         request.setAttribute("DVDList", dvdList);
24         rd.forward(request, response);
25     }
26
27     private List<DVDItem> createDvdList(){
28         List<DVDItem> dvds = new ArrayList<DVDItem>();
29         dvds.add(new DVDItem("Spiderman", "2000", "Action"));
30         dvds.add(new DVDItem("Forzen", "2014", "Animation"));
31         dvds.add(new DVDItem("Transformer4", "2014", "Action"));
32         return dvds;
33     }
34
35     HttpServlet methods. Click on the + sign on the left to edit the code.
36
73
74 }
```

list_library.jsp

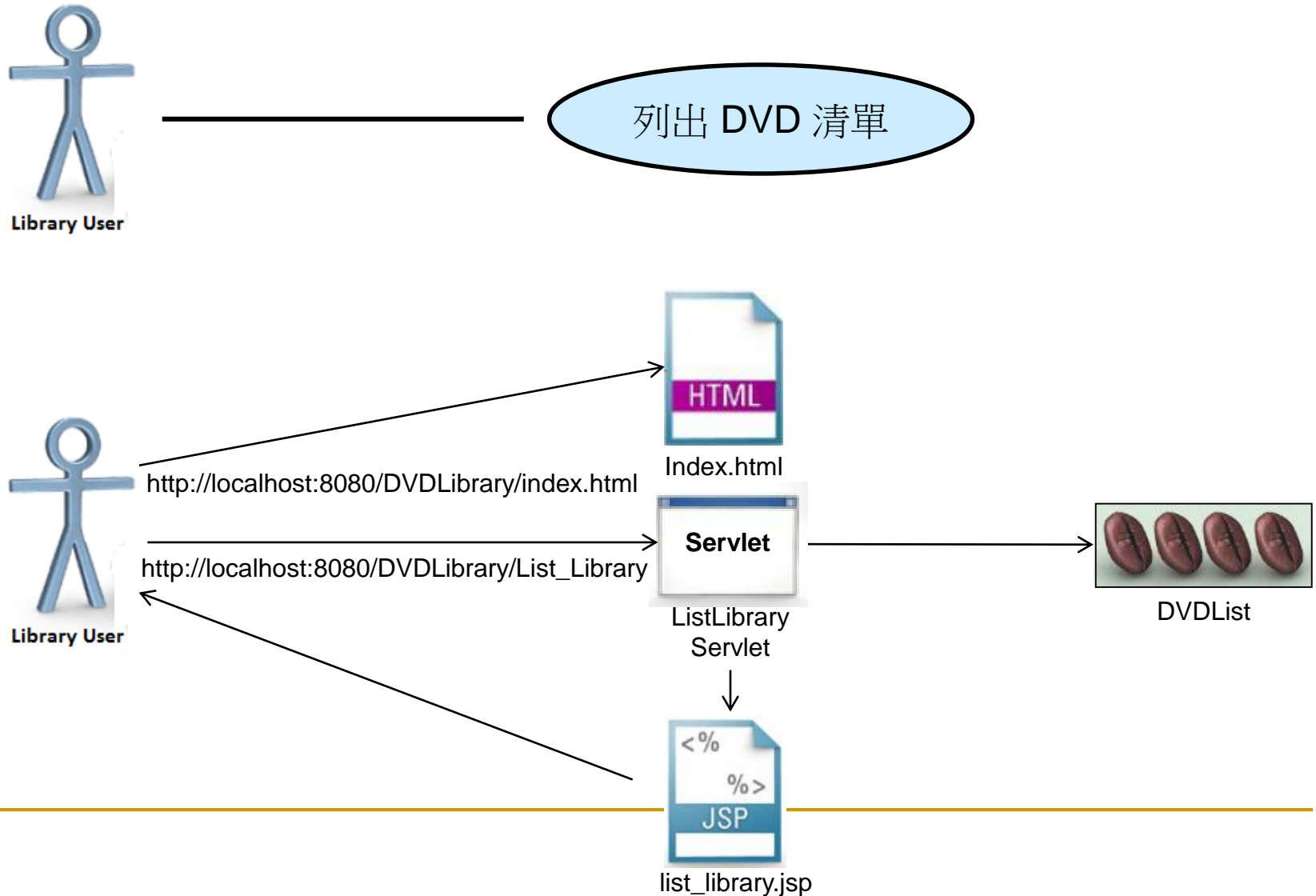


```
list_library.jsp x
Source History
<%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <title>DVD清單</title>
7   </head>
8   <body>
9     <h1>所有DVD:</h1>
10    ${DVDList}
11  </body>
12 </html>
```

測試、執行



Lab3 DVDLibrary 應用程式



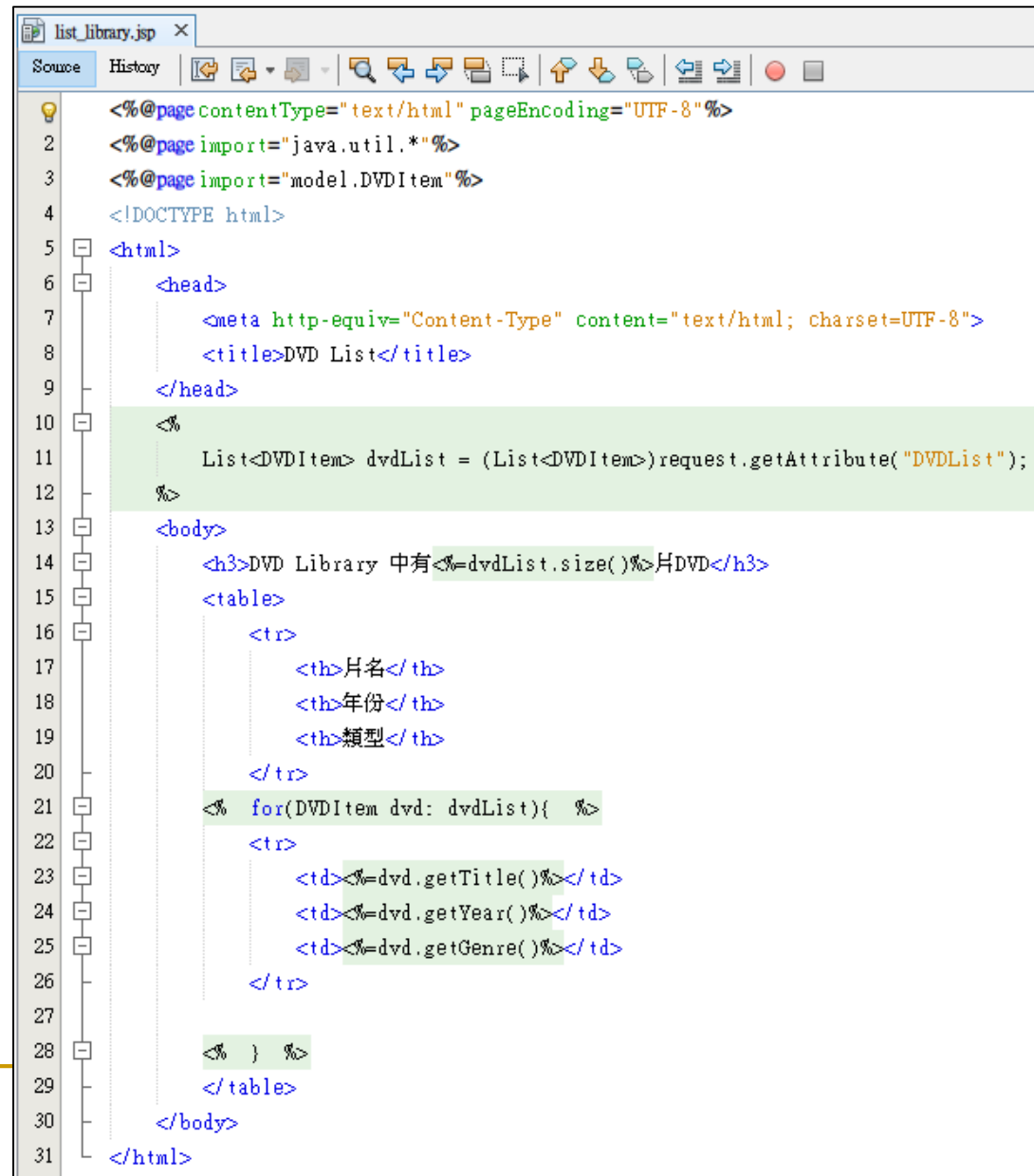
Lab3

- 改寫 list_library.jsp
 - 使用JSP scripting 元素
 - 取得dvdList屬性
 - 將DVD清單資訊顯示於網頁如下

- 測試、執行



list_library.jsp



```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <%@page import="java.util.*"%>
3 <%@page import="model.DVDItem"%>
4 <!DOCTYPE html>
5 <html>
6 <head>
7     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
8     <title>DVD List</title>
9 </head>
10 <%
11     List<DVDItem> dvdList = (List<DVDItem>)request.getAttribute("DVDList");
12 %>
13 <body>
14     <h3>DVD Library 中有<%=dvdList.size()%>片DVD</h3>
15     <table>
16     <tr>
17         <th>片名</th>
18         <th>年份</th>
19         <th>類型</th>
20     </tr>
21     <% for(DVDItem dvd: dvdList){ %>
22     <tr>
23         <td><%=dvd.getTitle()%></td>
24         <td><%=dvd.getYear()%></td>
25         <td><%=dvd.getGenre()%></td>
26     </tr>
27     <% } %>
28     </table>
29 </body>
30 </html>
```

測試、執行

