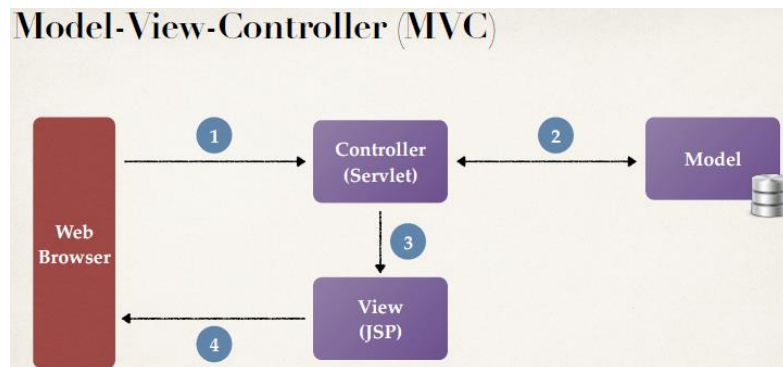


31_1_mvc_servlet: 서블릿과 JSP로 MVC 구현

베스트 프랙티스

- 서블릿과 JSP 둘을 통합!
- 서블릿은 비즈니스 로직 구현
- JSP는 프리젠테이션 뷰를 처리
- Model-View-Controller (MVC) 디자인 패턴



MVC의 이점

- 서블릿안에 있는 HTML 코드 최소화
 - 서블릿 코드내의 더 이상 `out.println(...)` 사용하지 않기
- JSP 내의 비즈니스 로직 최소화
 - 더 이상 JSP 코드내에 스크립트(scriptlet) 사용하지 않기

서블릿이 JSP를 호출하는 방법

- 서블릿은 리퀘스트 디스패처(*request dispatcher*)를 사용해 호출

// Step 1: request dispatcher 얻기

`RequestDispatcher dispatcher =`

`request.getRequestDispatcher("/view_students.jsp");`

// Step 2: JSP로 포워드(forward)

`dispatcher.forward(request, response);`

JSP로 데이터 보내는 방법

- 서블릿은 리퀘스트 오브젝트(request object) 데이터 추가

// Step 0: 데이터 추가

```
String[] students = {"Susan", "Anil", "Mohamed", "Trupti"};
```

```
request.setAttribute("student_list", students);
```

// Step 1: 리퀘스트 디스패처(request dispatcher) 얻기

```
RequestDispatcher dispatcher =
```

```
request.getRequestDispatcher("/view_students.jsp");
```

// Step 2: JSP로 포워드(forward)

```
dispatcher.forward(request, response);
```

수신 데이터 JSP 페이지 적용

- JSP는 데이터에 액세스 하기 위해 JSTL 사용

```
<%@ taglib uri = "http://java.sun.com/jsp/jstl/core" prefix = "c" %>
```

```
<html><body>
```

```
    <c:forEach var = "tempStudent" items = "${student_list}" >
```

```
        ${tempStudent} <br/>
```

```
    </c:forEach>
```

```
</body></html>
```

개발

- Step 1: 서블릿 클래스 만들기

The screenshot shows an IDE with a project structure on the left and a code editor on the right. The project structure includes a web application with various JSP and Java files. The code editor shows the implementation of the `MvcDemoServlet` class, which extends `HttpServlet`. The code includes package declarations, imports, and the `doGet` and `doPost` methods. Red boxes highlight specific parts of the code and project structure.

```
package com.javajspervlet_study7.servlet;

import java.io.IOException;

@WebServlet("/MvcDemoServlet")
public class MvcDemoServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public MvcDemoServlet() {
        super();
    }

    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        // Step 0: 데이터 추가
        String[] students = {"Susan", "Anil", "Mohamed", "Trupti"};
        request.setAttribute("student_list", students);

        // Step 1: request dispatcher 얻기
        RequestDispatcher dispatcher =
            request.getRequestDispatcher("/servlet/view_students.jsp");

        // Step 2: JSP로 request 포워드
        dispatcher.forward(request, response);
    }

    protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        doGet(request, response);
    }
}
```

- Step 2: 뷰 JSP 만들기

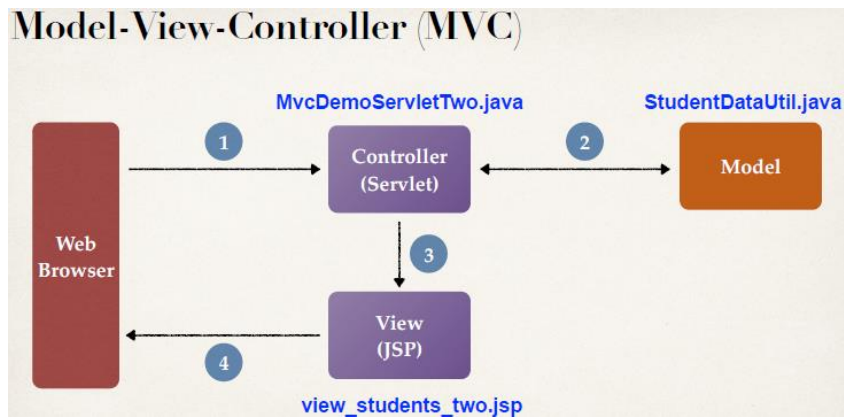
```

1  <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8" isELIgnored="false"%>
3  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
4  <!DOCTYPE html>
5  <html>
6  <head>
7     <meta charset="UTF-8">
8     <title>Insert title here</title>
9  </head>
10 <body>
11
12     <c:forEach var="student" items="${student_list}">
13         ${student} <br />
14     </c:forEach>
15
16 </body>
17 </html>

```

localhost:8080/javajspServlet_study7/MvcDemoServlet

32_1_mvc_with_tables



HTML Table

Student Table Demo

First Name	Last Name	Email
Mary	Public	mary@luv2code.com
John	Doe	john@luv2code.com
Ajay	Rao	ajay@luv2code.com

할 일

- Student 클래스 만들기
- StudentDataUtil 클래스 만들기
- MVC 서블릿 만들기
- View JSP 만들기

개발

- Step 1: 모델(Model) 클래스 만들기

```
Student.java StudentDataUtil.java
1 package com.javajspServlet_study7.servlet.mvc;
2
3 public class Student {
4
5     private String firstName;
6     private String lastName;
7     private String email;
8
9     public Student(String firstName, String lastName, String email) {
10         this.firstName = firstName;
11         this.lastName = lastName;
12         this.email = email;
13     }
14
15     public String getFirstName() {
16         return firstName;
17     }
18
19     public void setFirstName(String firstName) {
20         this.firstName = firstName;
21     }
22
23     public String getLastName() {
24         return lastName;
25     }
26
27     public void setLastName(String lastName) {
28         this.lastName = lastName;
29     }
30
31     public String getEmail() {
32         return email;
33     }
34
35     public void setEmail(String email) {
36         this.email = email;
37     }
38 }
```

- Step 2: StudentDataUtil 클래스 만들기

```
Student.java StudentDataUtil.java
1 package com.javajspServlet_study7.servlet.mvc;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class StudentDataUtil {
7
8     public static List<Student> getStudents()
9     {
10         // 빈 리스트 만들기
11         List<Student> students = new ArrayList<Student>();
12
13         // 샘플 데이터 추가
14         students.add(new Student("k1", "k1@www.com"));
15         students.add(new Student("k2", "k2@www.com"));
16         students.add(new Student("k3", "k3@www.com"));
17
18         // 리스트 리턴
19         return students;
20     }
21
22 }
```

- Step 3: MVC 서블릿 만들기

```

MvcDemoServlet2.java StudentDataUtil.java view_students2.jsp
1 package com.javajspServlet_study7.servlet.mvc;
2
3 import java.io.IOException;
4
5
6
7
8
9
10
11
12
13
14 @WebServlet("/MvcDemoServlet2")
15 public class MvcDemoServlet2 extends HttpServlet {
16     private static final long serialVersionUID = 1L;
17
18     public MvcDemoServlet2() {
19         super();
20     }
21
22     protected void doGet(HttpServletRequest request,
23         HttpServletResponse response)
24         throws ServletException, IOException {
25         // Step 1: 헬퍼 클래스(모델)로부터 학생을 가져옴
26         List<Student> students = StudentDataUtil.getStudents();
27
28         // Step 2: request object에 학생 추가
29         request.setAttribute("student_list", students);
30
31         // Step 3: request dispatcher 가져오기
32         RequestDispatcher dispatcher =
33             request.getRequestDispatcher("/servlet/view_student2.jsp");
34
35         // Step 4: JSP로 포워드
36         dispatcher.forward(request, response);
37     }
38
39     protected void doPost(HttpServletRequest request,
40         HttpServletResponse response)
41         throws ServletException, IOException {
42         doGet(request, response);
43     }
44 }

```

- Step 4: 뷰 JSP 만들기

```

MvcDemoServlet2.java StudentDataUtil.java view_students2.jsp
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2     pageEncoding="UTF-8" isELIgnored="false"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
4 <!DOCTYPE html>
5 <html>
6 <head>
7     <meta charset="UTF-8">
8     <title>Insert title here</title>
9 </head>
10 <body>
11     <h2>학생 테이블</h2>
12     <hr>
13     <table border="1">
14         <tr>
15             <th>이름</th>
16             <th>성</th>
17             <th>이메일</th>
18         </tr>
19         <c:forEach var="student" items="${student_list}">
20             <tr>
21                 <td>${student.firstName}</td> Student.getFirstName() 함수 호출
22                 <td>${student.lastName}</td> Student.getLastName() 함수 호출
23                 <td>${student.email}</td> Student.getEmail() 함수 호출
24             </tr>
25         </c:forEach>
26     </table>
27 </body>
28 </html>

```