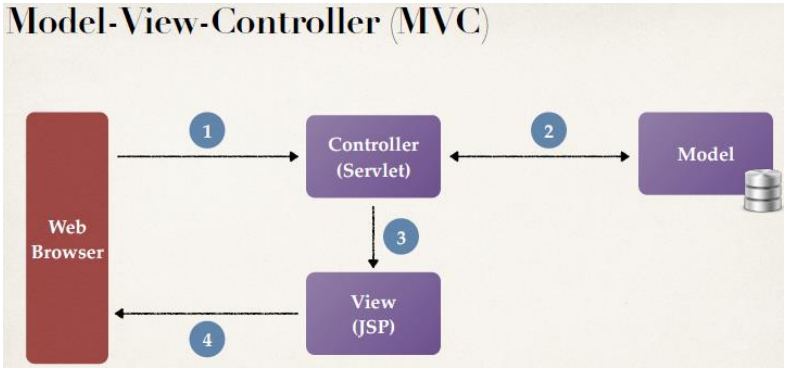
31_1_mvc_servlet: 서블릿과 JSP로 MVC 구현

베스트 프렉티스

- 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 스클립트(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);
```

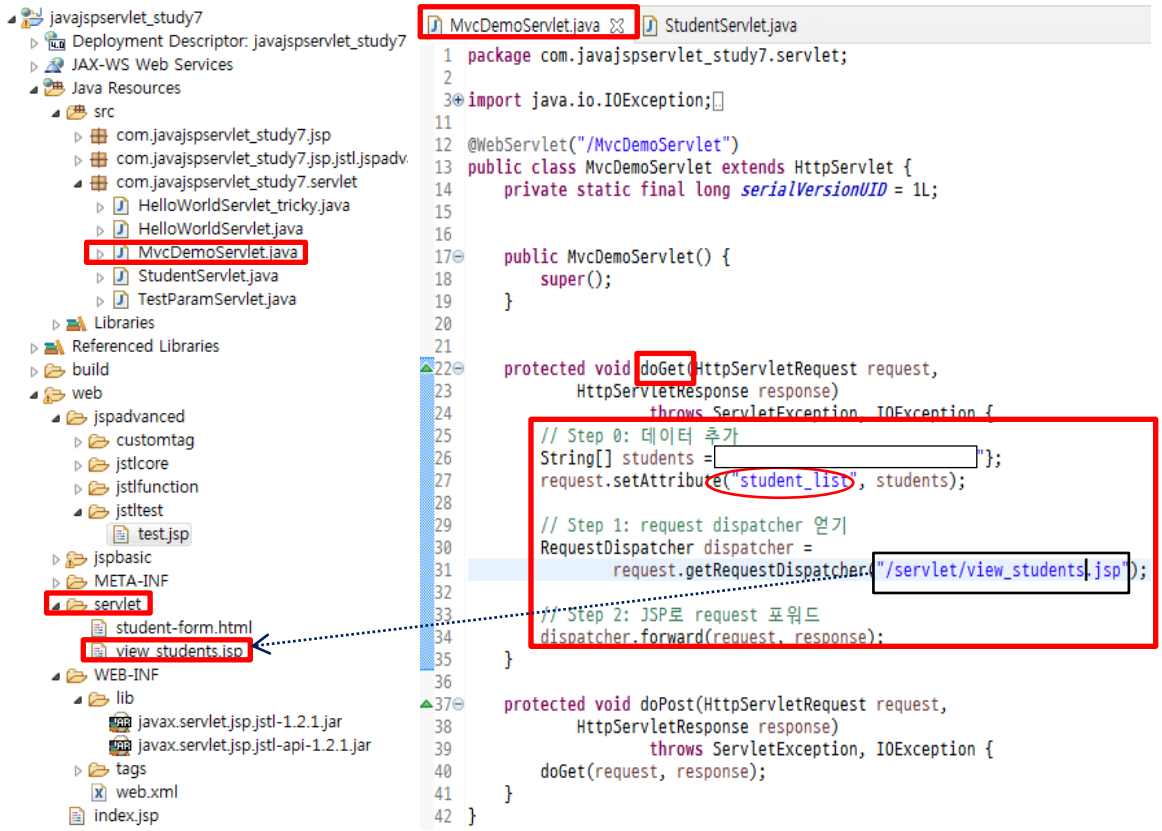
수신 데이터 JSP 페이지 적용

- JSP는 데이터에 엑세스 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

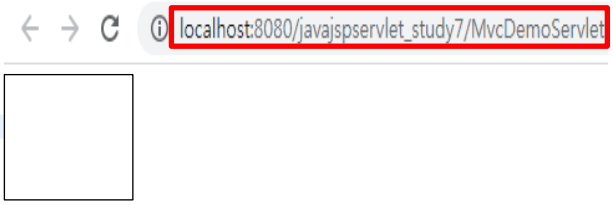
- Step 1: 서블릿 클래스 만들기



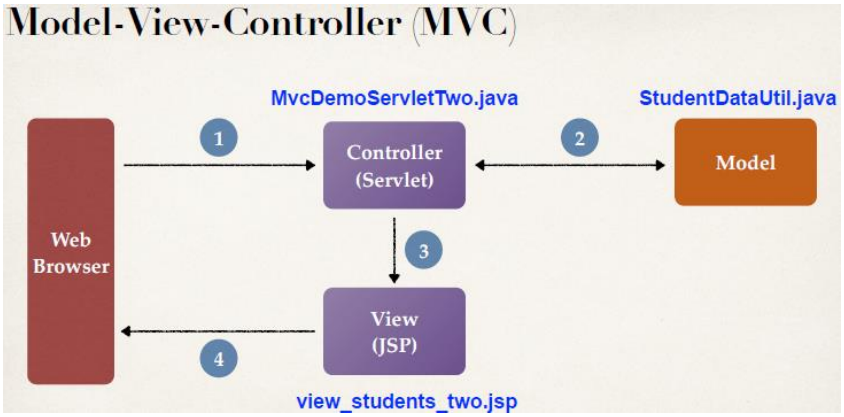
- Step 2: 뷰 JSP 만들기

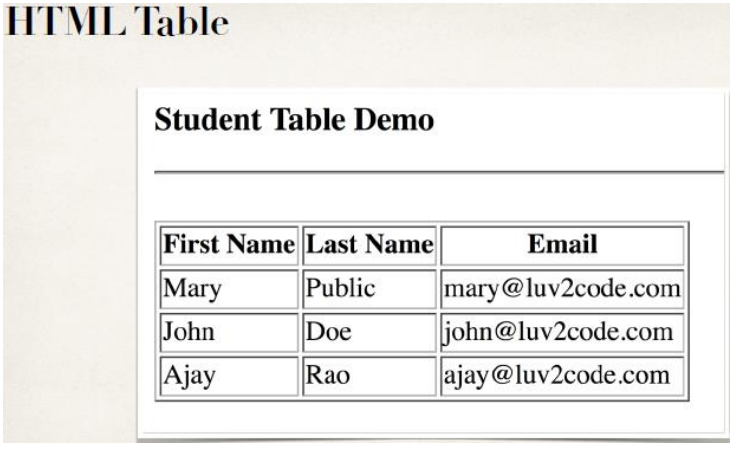
MvcDemoServlet.java | view_students.jsp

```
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
12         <c:forEach var="student"  items="${student_list}">
13
14             ${student} <br />
15
16         </c:forEach>
17
18     </body>
19 </html>
```



32_1_mvc_with_tables





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

```java
package com.javajspservlet_study7.servlet.mvc;

public class Student {

    private String firstName;
    private String lastName;
    private String email;

    public Student(String firstName, String lastName, String email) {
        this.firstName = firstName;
        this.lastName = lastName;
        this.email = email;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }
}
```

- Step 2: StudentDataUtil 클래스 만들기

Student.java | StudentDataUtil.java

```java
package com.javajspservlet_study7.servlet.mvc;

import java.util.ArrayList;
import java.util.List;

public class StudentDataUtil {

    public static List<Student> getStudents()
    {
        // 빈 리스트 만들기
        List<Student> students = new ArrayList<Student>();

        // 샘플 데이터 추가
        students.add(new Student(                , "k1@www.com"));
        students.add(new Student(                , "k2@www.com"));
        students.add(new Student(                , "k3@www.com"));

        // 리스트 리턴
        return students;
    }

}
```

- Step 3: MVC 서블릿 만들기

MvcDemoServlet2.java | StudentDataUtil.java | view_students2.jsp

```java
package com.javajspservlet_study7.servlet.mvc;

import java.io.IOException;


@WebServlet("/MvcDemoServlet2")
public class MvcDemoServlet2 extends HttpServlet {
    private static final long serialVersionUID = 1L;

    public MvcDemoServlet2() {
        super();
    }

    protected void doGet(HttpServletRequest request,
            HttpServletResponse response)
                    throws ServletException, IOException {
        // Step 1: 헬퍼 클래스(모델)로부터 학생을 가져옴
        List<Student> students = StudentDataUtil.getStudents();

        // Step 2: request object에 학생 추가
        request.setAttribute("student_list", students);

        // Step 3: request dispatcher 가져오기
        RequestDispatcher dispatcher =
                request.getRequestDispatcher("/servlet/view_student2.jsp");

        // Step 4: JSP로 포워드
        dispatcher.forward(request, response);
    }

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

- Step 4: 뷰 JSP 만들기

MvcDemoServlet2.java | StudentDataUtil.java | view_students2.jsp

```jsp
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" isELIgnored="false"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>Insert title here</title>
    </head>
    <body>
        <h2>학생 테이블</h2>
        <hr>
        <table border="1">
            <tr>
                <th>이름</th>
                <th>성</th>
                <th>이메일</th>
            </tr>
            <c:forEach var="student" items="${student_list}">
                <tr>
                    <td>${student.firstName}</td>
                    <td>${student.lastName}</td>
                    <td>${student.email}</td>
                </tr>
            </c:forEach>
        </table>
    </body>
</html>
```

Student.getFirstName() 함수 호출
Student.getLastName() 함수 호출
Student.getEmail() 함수 호출