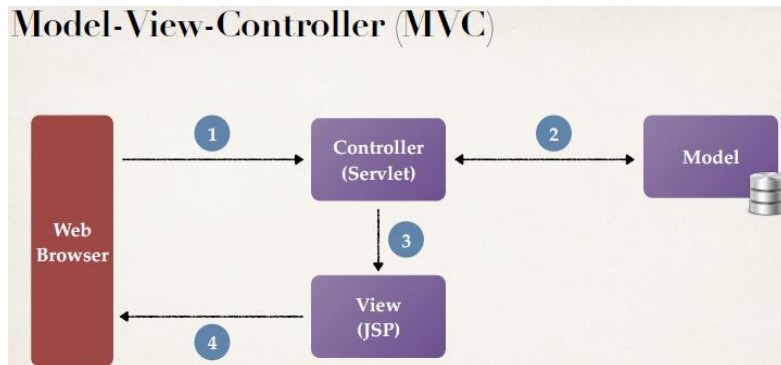


1. MVC 서블릿: 서블릿과 JSP로 MVC 구현

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

- 서버릿은 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/>
```

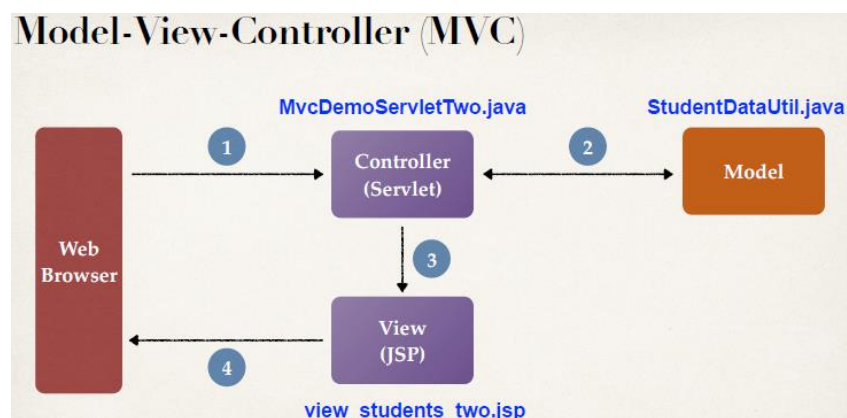
```
    </c:forEach>
```

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

실습1

- Step 1: 서버릿 클래스 만들기
- Step 2: 뷰 JSP 만들기

2. 테이블로 작업하기



할 일

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

실습

- Step 1: 모델(Model) 클래스 만들기
- Step 2: StudentDataUtil 클래스 만들기
- Step 3: MVC 서블릿 만들기
- Step 4: 뷰 JSP 만들기