

Class 230112

CGI (common gateway interface, Perl) → Servlet → JSP → Spring (Spring Boot)

Servlet

장점

1. JAVA 기반으로 일반 JAVA의 모든 API들을 그대로 사용할 수 있다.
2. 한번 개발 된 애플리케이션은 다양한 서버의 환경에서 실행할 수 있다.
3. 다양한 오픈소스 라이브러리들과 개발 도구들을 활용할 수 있다.

단점

1. HTML 출력을 위해서 문자열 결합을 복잡하게 해야 한다.
2. 프론트엔드 화면 수정이 아주 어렵다.
3. 데이터 처리가 불편하다.

▼ 전형적인 sevlet code 예시

```
@WebServlet(name = "memberSaveServlet", urlPatterns = "/servlet/members/save")
public class MemberSaveServlet extends HttpServlet {

    private final MemberRepository memberRepository = MemberRepository.getInstance();

    @Override
    protected void service(HttpServletRequest request, HttpServletResponse response) {
        final String username = request.getParameter("username");
        final int age = Integer.parseInt(request.getParameter("age"));
        final Member member = new Member(username, age);
        memberRepository.save(member);

        response.setContentType("text/html");
        response.setCharacterEncoding("utf-8");
        final PrintWriter printWriter = response.getWriter();
        printWriter.write("<html>\n" +
            "<head>\n" +
            " <meta charset=\"UTF-8\">\n" +
            "</head>\n" +
            "<body>\n" +
            "성공\n" +
            "<ul>\n" +
            " <li>id=" + member.getId() + "</li>\n" +
            " <li>username=" + member.getUsername() + "</li>\n" +
            " <li>age=" + member.getAge() + "</li>\n" +
            "</ul>\n" +
            "<a href=\"/index.html\">메인</a>\n" +
            "</body>\n" +
            "</html>");
    }
}
```

▼ 전형적인 JSP code 예시

```

<%@ page import="hello.servlet.basic.domain.member.Member" %>
<%@ page import="hello.servlet.basic.domain.member.MemberRepository" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<%
    final MemberRepository memberRepository = MemberRepository.getInstance();
    final String username = request.getParameter("username");
    final int age = Integer.parseInt(request.getParameter("age"));
    final Member member = new Member(username, age);
    memberRepository.save(member);
%>
<html>
<head>
    <title>Title</title>
</head>
<body>
    성공
    <ul>
        <li>id=<%=member.getId()%>
        </li>
        <li>username=<%=member.getUsername()%>
        </li>
        <li>age=<%=member.getAge()%>
        </li>
    </ul>
    <a href="/index.html">메인</a>
</body>
</html>

```

서블릿은 SERVER APPLET의 줄임말로 cgi의 단점을 보완하기 위해서 썬마이크로시스템사(지금은 오라클로 흡수)에서 개발한 동적 웹 솔루션이다. 서블릿을 기반으로 하여 JSP가 개발되었다.

JSP는 내부적으로 서블릿으로 변환되어 실행된다. 따라서 서블릿의 구성이나 동작 방식을 이해하면 JSP를 이해하는데 상당한 도움이 될 것이다.

비교항목	서블릿	JSP
프로그래밍 스타일	<div>Java 코드</div> <div>HTML 코드</div>	<div>HTML 코드</div> <div>Java 코드</div>

실습

form 태그로 호출한 데이터를 서블릿 페이지로 전송하여 처리 후 화면에 출력한다.

1. signup.html 에서 입력 받은 데이터를 SignupServlet에서 처리하여 화면 출력

▼ 화면

회원가입

아이디

패스워드

패스워드(확인)

이름

User ID: Kim

Password: 111

Name: Simin Kim

▼ signup.html

```
<html>
<head>
<meta charset="utf-8">
</head>
<body>
<form action="SignupServlet" method="post">
<table align=center>
<tr><td colspan=2 align=center height=40><b>회원가입</b></td></tr>
<tr>
<td align=right>아이디 </td>
<td><input type="text" name="id" placeholder="Email address" required></td>
</tr>
<tr>
<td align=right>패스워드 </td>
<td><input type="password" name="ps" required></td>
</tr>
<tr>
<td align=right>패스워드(확인) </td>
<td><input type="password" name="ps2" required></td>
</tr>
<tr>
<td align=right>이름 </td>
<td><input type="text" name="name" required></td>
</tr>
<tr>
<td colspan=2 align=center height=50>
<input type="submit" value="회원가입하기">
</td>
</tr>
</table>
</form>
```

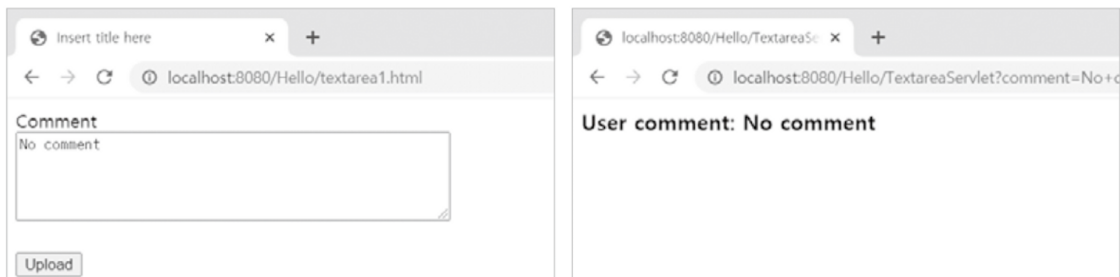
```
</body>
</html>
```

▼ SignupServlet

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // POST로 보내더라도 doGet 메서드에서 처리한다.
    String res = "<html><h3>User ID : " + request.getParameter("id") + "</h3>";
    res += "<h3>Password : " + request.getParameter("ps") + "</h3>";
    res += "<h3>Name : " + request.getParameter("name") + "</h3></html>";
    response.getWriter().print(res);
}
```

2. textarea로 입력 받은 데이터를 서블릿에서 처리하여 화면 출력

▼ 화면



▼ textarea1.html

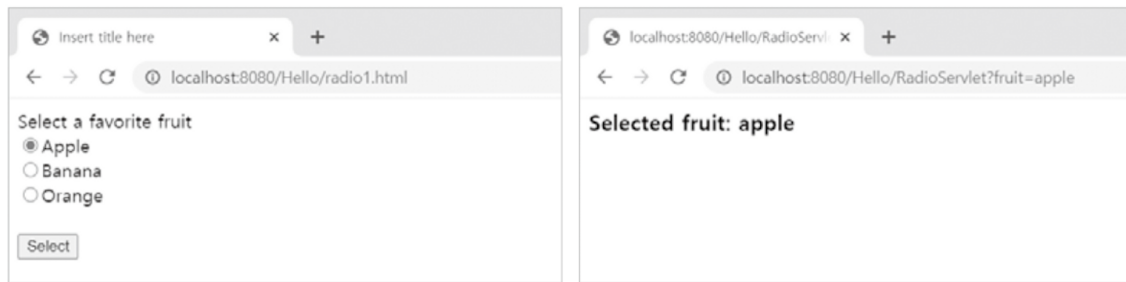
```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<form action="TextareaServlet">
<span>Comment</span><br>
<textarea name="comment"></textarea><br><br>
<input type="submit" value="Upload">
</form>
</body>
</html>
```

▼ TextareaServlet

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // POST로 보내더라도 doGet 메서드에서 처리한다.
    String res = "<html><h3>User ID : " + request.getParameter("id") + "</h3>";
    res += "<h3>Password : " + request.getParameter("ps") + "</h3>";
    res += "<h3>Name : " + request.getParameter("name") + "</h3></html>";
    response.getWriter().print(res);
}
```

3. 라디오 버튼으로 입력 받은 데이터를 서블릿에서 처리하여 화면 출력

▼ 화면



▼ radio1.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
  <form action="RadioServlet"> <!-- 입력한(선택한) 값을 넘겨줄 서블릿 페이지 -->
    <span>Select a favorite fruit</span><br>
    <input type="radio" name="fruit" value="apple">Apple<br>
    <input type="radio" name="fruit" value="banana">Banana<br>
    <input type="radio" name="fruit" value="orange">Orange<br>
    <br>
    <input type="submit" value="Select">
  </form>
</body>
</html>
```

▼ RadioServlet

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String res = "<html><h3>Selected fruit : " + request.getParameter("fruit") + "</h3>";
    response.getWriter().print(res);
}
```