

Class 230112

Hour 1

about

Hour 2

check

Hour 3

env 1

Hour 4

env 2

Hour 5

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

Servlet 장점

- 자바 기반이기 때문에 일반 자바의 모든 api 들을 그대로 가져다가 쓸수 있다.
- 한번 개발된 애플리케이션은 다양한 서버 환경에서 실행할수 있다.
- 다양한 오픈소스 라이브러리들과 개발도구들을 활용할수 있다.

단점

- HTML 출력을 위해서 문자열 결합을 복잡하게 해야 한다.
- 프론트엔드 화면 수정이 아주 어렵다.
- 데이터 처리도 불편하다.

전형적인 SERVLET

```

@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

```

<%@ 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 APPLET 의 줄임말로 cgi의 단점을 보완하기 위해서 썬마이크로시스템사(지금은 오라클로 흡수)에서 개발한 동적 웹 솔루션이다. 서블릿을 기반으로 하여 JSP 가 개발되었다.

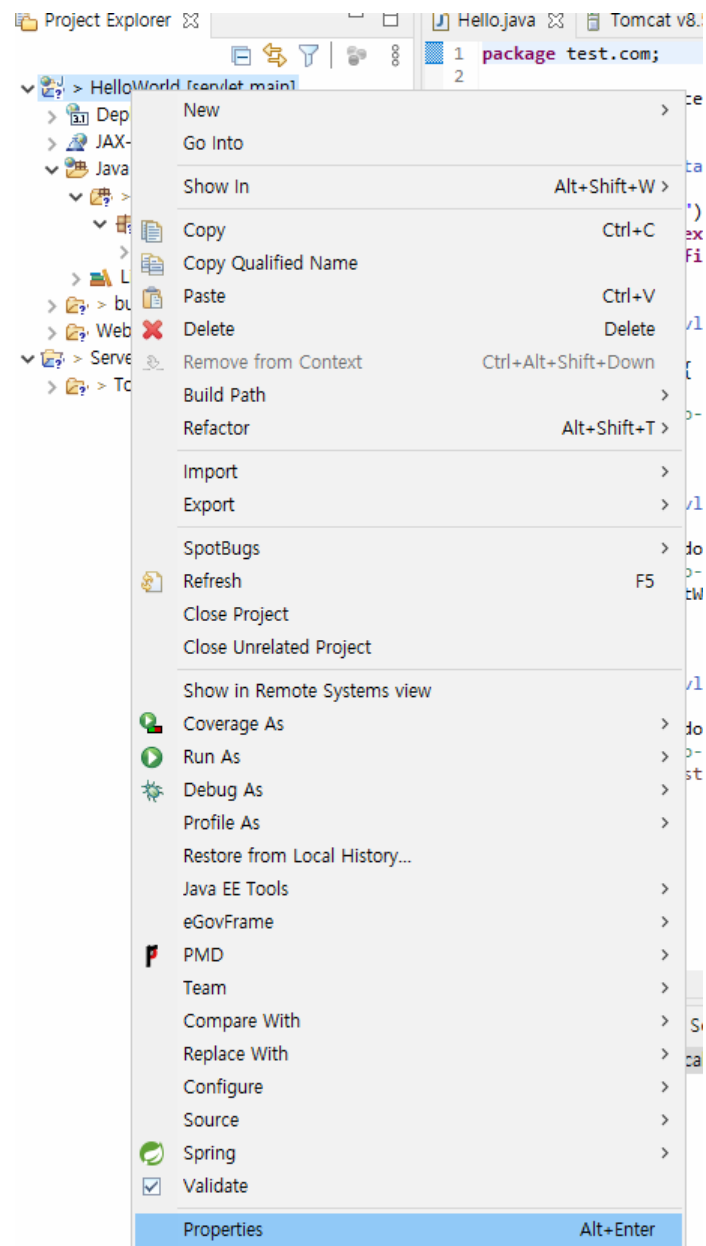
JSP 는 내부적으로 서블릿으로 변환되어 실행된다. 따라서 서블릿의 구성이나 동작 방식을 이해하면 JSP 를 이해하는데 상당한 도움이 될것이다.

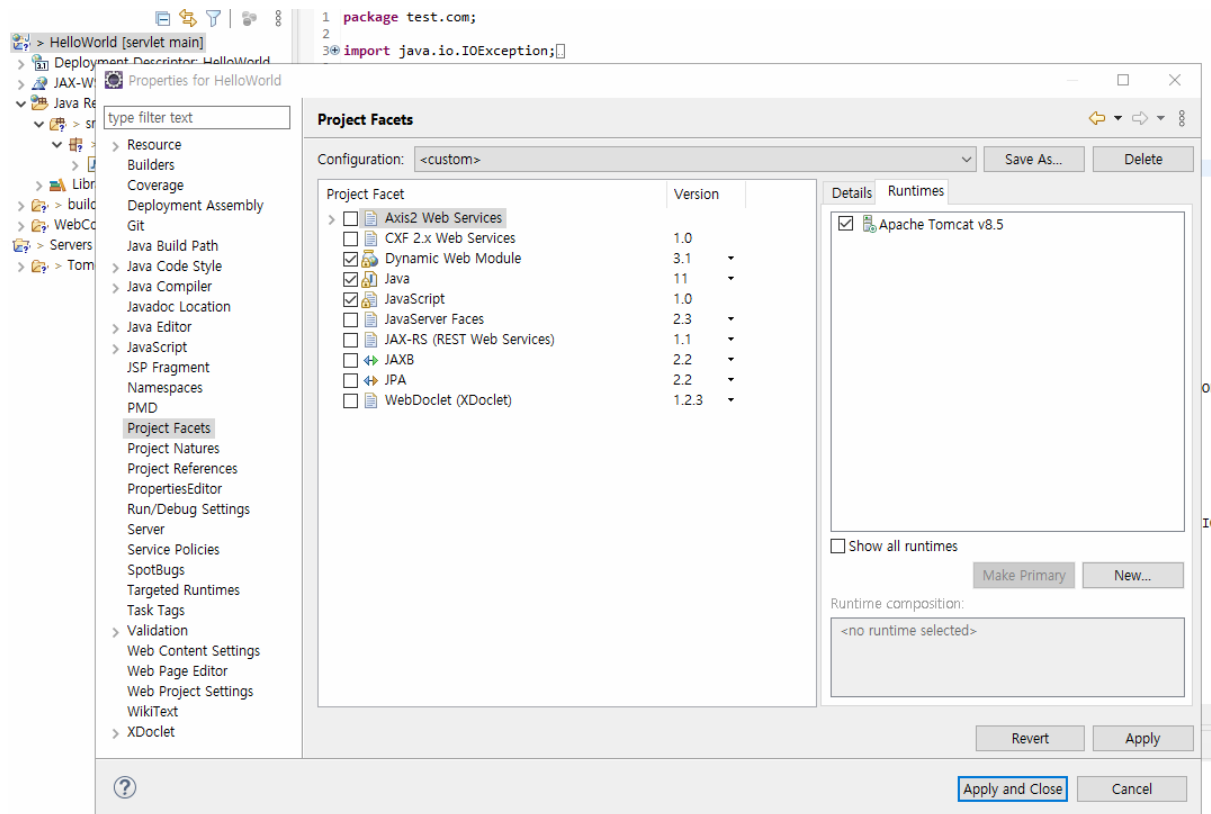
비교항목	서블릿	JSP
프로그래밍 스타일	<div>Java 코드</div> <div>HTML 코드</div>	<div>HTML 코드</div> <div>Java 코드</div>

<https://youtu.be/rZlKTcmILSo>

- 처음 설치후 실행시 발생한 문제점들...

- 순서를 서버부터 설치하고 tomcat 8.5 그리고 프로젝트 시작
-





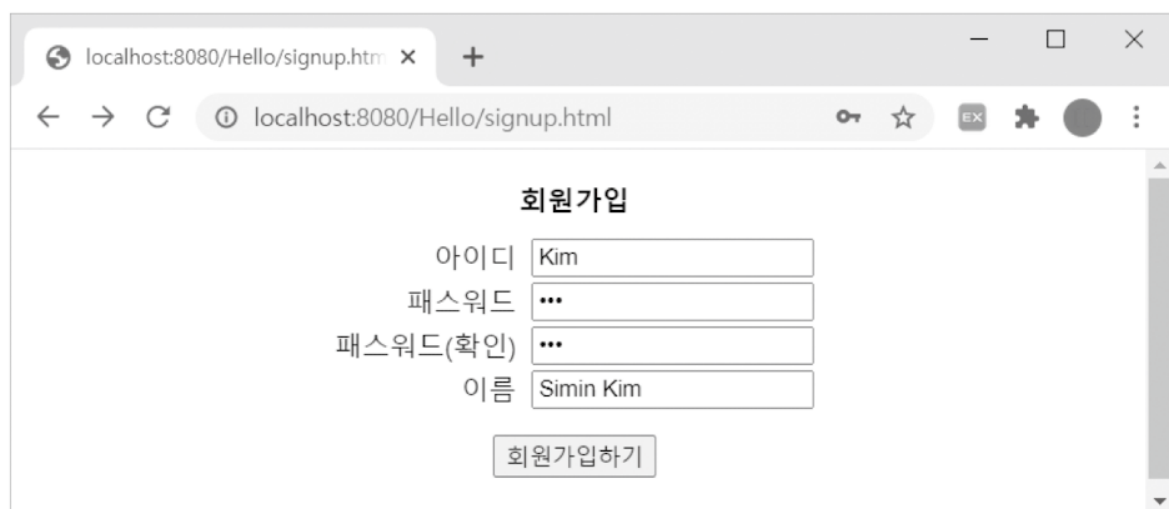
runtime 에 tomcat 체크

Hour 6

Servlet 2

Hour 7

Servlet 3

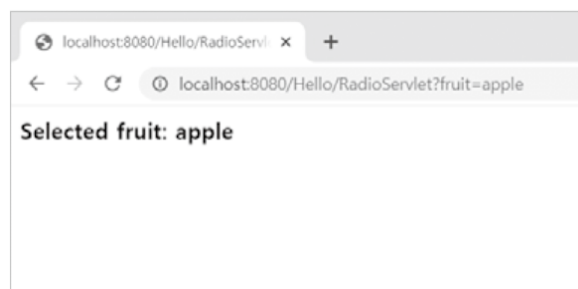
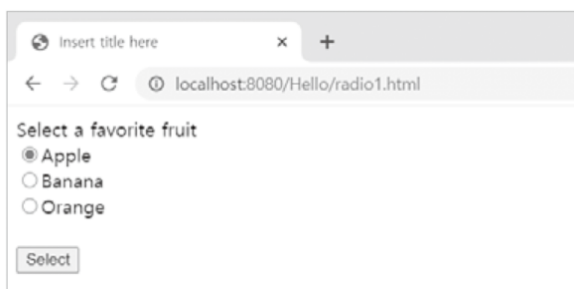
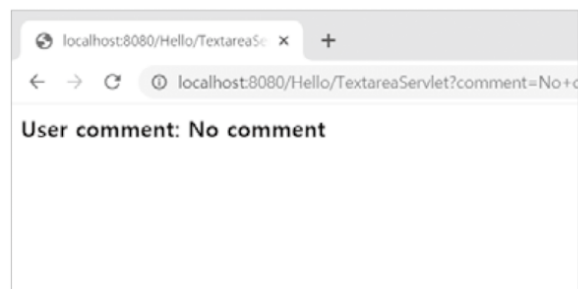
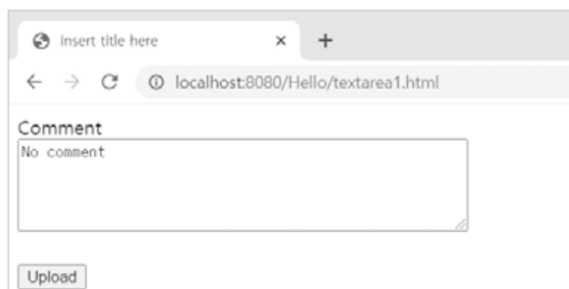




```
String res = "<html><h3>User ID: " + request.getParameter("id") + "</h3>";
res += "<h3>Password: " + request.getParameter("ps") + "</h3>";
res += "<h3>Name: " + request.getParameter("name") + "</h3></html>";
response.getWriter().print(res);
```

```
<form action="SignupServlet" method="post">
```

form 태그로 호출한 서블릿 페이지로 데이터를 전송하여 서블릿을 통해 화면에 출력한다.



```

<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
<form action="RadioServlet">    <!-- 입력한(선택한) 값을 넘겨줄 서블릿 페이지 호출
Select a favorite fruit <br>
<input type="radio" name="fruit" value="apple" checked>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>

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    String use1 = request.getParameter("fruit");
    String res = "<html><h3>Selected fruit: " + use1 + "</h3></html>";
    response.getWriter().print(res);
}

```

Hour 8

Github & Notion

Question

Please sign in

Email address

Password

☐ Remember me

Sign in

공통 : Open Book, Open Internet, Open Source ...

Email address 칸에 abc@naver.com 와 Password 0000을 입력 후 Sign in 단추를 누를 수 있는 login.jsp

입력한 값 비교 처리를 위한 logincheck.jsp

정확히 입력한 경우 “로그인 성공”이라는 글자를 출력하는 ok.jsp

다를 경우 “로그인 실패”라는 글자를 출력하는 no.jsp

네 개의 페이지를 로컬 웹서버에서 동작하도록 만드세요. (기본은 DB없이 logincheck.jsp 파일에서 변수로 비교) 디자인은 전혀 없이 동작만 해도 됩니다.

추가 1 : DB에 member 테이블을 만들어 조회 후 동작하도록 만드세요.(oracle, mysql, maria 등등 원하는 db 사용가능, backend 확인용)

추가 2 : 위 그림과 거의 동일하거나 다른 형태로 꾸미세요 (frontend 확인용)

추가 3 : 로그인 성공 시 세션 처리

공통 + 추가 1 + 추가 2 + 추가 3 모두 완성 시 선착순 상품 지급

제출은 login.jsp, logincheck.jsp, ok.jsp, no.jsp 4개 파일을 “본인이름.zip”로 압축하여 austiny@snu.ac.kr 로 제출
→ 채팅 창에 제출