

Class 230112 (서블릿)

SERVELT 서블릿

웹 개발의 흐름(간단)

CGI(Common) → Servlet → JSP → Spring (Spring Boot)

Servlet에 관하여

장점

- 자바 기반 이므로 자바의 모든 api들을 그대로 가져다가 쓸 수 있음
- 한번 개발된 애플리케이션은 다양한 서버 환경 에서 실행 가능
- 다양한 오픈소스 라이브러리들과 개발도구들을 활용할 수 있음

단점

- HTML 출력을 위해서는 복잡하게스리.. 문자열 결합 을 해야 함
- 프론트엔드 화면 수정이 아주 어려움
- 데이터 처리도 불편함

개념

- 서블릿은 SERVER APPLET 의 줄임말
- cgi의 단점을 보완하기 위해서 썬마이크로시스템사(지금은 오라클로 흡수)에서 개발한 동적 웹 솔루션
- 서블릿을 기반으로 하여 JSP 가 개발됨
- 우리가 JSP 를 써도 내부적으로 Servlet 으로 바뀌어서 돌아감
- 즉 서블릿의 구성 혹은 동작 방식을 이해하면 JSP를 이해하기도 좋음

서블릿으로 화면 그리기 실습

```
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html;charset=utf-8");
    PrintWriter out = response.getWriter();
    out.append("<!doctype html><html><head><title>Hello World Servlet</title></head><body>")
    .append("UserID : " + request.getParameter("id"))
    .append("<br>Password : " + request.getParameter("ps"))
    .append("<br>Name : " + request.getParameter("name"))
    .append("</body></html>");
}
```

`request.getParameter()` 함수를 통해 넘겨진 파라미터를 받음.

`response.getWriter()` 함수는 `PrintWriter` 객체를 반환

`out` 객체에 `append()` 한 문자열은 화면에 나옴

```
String res = "<html><h3>USER COMMENTS: " + request.getParameter("comment") + "</h3></html>";
response.getWriter().print(res);
```

이 경우에 `.print()` 함수에 매개변수로 넣어준 문자열(HTML)이 화면에 뜸