

JSP

[03] Servlet 맛보기I <Servlet은 java파일>

1. Servlet 문서 작성하기

(1) Servlet의 특징

- ① 동적 웹어플리케이션 컴포넌트 ; JSP랑 같음
- ② .java 확장자 ; JAVA 언어를 사용하여 웹 프로그램을 제작하는 것

Servlet (Interface)
↑ (구현)
GenericServlet (abstract)
↑ (상속)
HttpServlet
↑ (상속)

우리가 작성한 서블릿 클래스

- ③ 클라이언트의 요청에 동적으로 작동하고, 응답은 html을 이용
- ④ **java thread**이용하여 동작 ; 서블릿을 이용하는 가장 큰 장점. 웹컴포넌트를 구성하는 다른 언어에 비해, 요청이 많이 들어와도 서버의 부하가 많이 걸리지 않는 장점
 - **thread가 필요한 이유** : 채팅이나 영화관 사이트는 여러 사람이 한번에 동시에 접속, 동시에 접속했을 경우 약간의 차이를 두고 실행시키며 클라인트 입장에서 동시에 실행하는 것 처럼 보여야 하니까!
- ⑤ MVC패턴에서 Controller 부분에 이용됨

```
<body>
  <button onclick="location.href='HW'">HelloWorld서블릿 요청</button>
  <!-- HW라는 url을 매핑시키는 화일(자바파일)을 만들어야 함 = 서블릿(컨트롤러를 만드는 재료가 되는 아이) -->
  <!-- Jsp로 요청할경우(가려고 할 경우) -->
  <!-- <button onclick = "location.href='hello.jsp'">Hello World</button> -->
</body>
</html>
```

자바를 이용하여 서블릿 요청하는 방법과 j네를 요청하는 방법이 좀다름

단 자바는 Java Resources 의 src에 만들어야됨 서블릿의 형태로!!!

서블릿만들 때 package : com.lec.ex / class name : Ex1_HelloWold 하고 **절대 finish 누르지 말고 next를 누른 후 매핑작업실행**

Mapping pattern 은 대소문자 구분함 / 도 필수

JSP 는 알아서 실행해주고 알아서 변환시켜주지만 서블릿은 eclipse가 컴파일 후 톰캣이 가져가야

함 JSP 는 수정후 다시 실행하면 알아서 수정한게 실행되지만 **서블릿은 내용을 수정하면 톰캣을 다시 재실행해야 먹힘 (갓다 키든 restart를 하던)**

```
@WebServlet(name = "HelloWorld", urlPatterns = { "/HWorld" })  
//url 매핑은 필수
```

서블릿은 url 매핑을 꼭 해줘야됨 안해주면 안됨

```
* @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse response)  
*/  
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws Ser  
// TODO Auto-generated method stub  
response.getWriter().append("Served at: ").append(request.getContextPath());  
}
```

브라우저에서 url 달라고 하면 그냥 주는게 아니라 요청객체를 생성후 들어옴

요청객체를 request 안에 파라미터로 넣어줌

그럼 객체가 request에 달라붙어 파라미터도 달라붙음

따라서 파일만 요청하면 안되고, query까지 같이 요청하는 경우가 많음

Request가 id 가 뭐구나, pw가 뭐구나 확인후 데이터 베이스를 확인함

결과값에 대한 페이지를 톰캣이 만들어 브라우저에 씀

JSP 페이지를 만들면 자바파일로 바뀜 하지만 서블릿은?

Response 객체에 결과값을 HTML 형태로 심어서 그 객체를 다시 내보내줌

<파일 입출력시 String(빨대형태) 쓰는거랑 비슷

Response.getWriter().append("served at"); ----브라우저에 뿌리기

```
37 */  
38 protected void doPost(HttpServletRequest request, HttpServletResponse response) throws Ser  
39     System.out.println("doPost호출");  
40     doGet(request, response);  
41     //method 불러올때 post 쓰면 애까 불러와지고  
42     //로그인이나 게시판은 post 방식으로 호출해야 됨  
43 }
```

정보: 프로토콜 핸들러 ["http-nio-8090"]을(를) 시작합니다.
5월 20, 2022 10:57:13 오전 org.apache.catalina.startup.Catalina
정보: 서버가 [478] 밀리초 내에 시작되었습니다.
doPost호출
doGet호출

첫번째 캡처본의 doGet 이 위단에 쓴 protected void doGet을 불러와서 doGet 호출이가능

** 어노테이션을 이용한 서블릿 맵핑

1. `@WebServlet(name = "HelloWorld", description = "Ex1", urlPatterns = { "/HelloWorld" })` (class 이름과 url 이름이 다를 때 사용 or 같을때도 사용 가능)
2. `@WebServlet("/HelloWorld")` (class 이름과 url 이름이 같을 때만 사용 가능)
3. 어노테이션을 지우고 url 맵핑을 web.xml에서도 할 수 있음

```
<?xml version="1.0" encoding="UTF-8"?>
<servlet>
  <servlet-name>HelloWorld</servlet-name>
  <servlet-class>com.lec.ex.Ex1_HelloWorld</servlet-class>
</servlet>

<servlet-mapping>
  <servlet-name>HelloWorld</servlet-name>
  <url-pattern>/hw</url-pattern>
</servlet-mapping>
</web-app>
```

webContent/ex1.html

Ex1

애네는 프로젝트 바로 밑에 있으니 같은 경로 느낌

같은 프로젝트 밑이니 바로 어세스가 가능함

<http://localhost:8090/pro/ex1.html>

<http://localhost:8090/pro/Ex1>

EX3을 어세스 하고 싶으면?(webContent/ex/ex3.html) 다른 폴더

<http://localhost:8090/pro/ex/ex3.html>

애는 프로젝트 밑에 ex 밑에 ex3이 있기 때문에 ex1.html과 폴더가 다름

따라서 ex3과 EX1은 다른 폴더이기 때문에 바로 어세스가 불가능