

[0630]SERVLET+SCOPE

SERVLET

- multiple context오류가 났을때

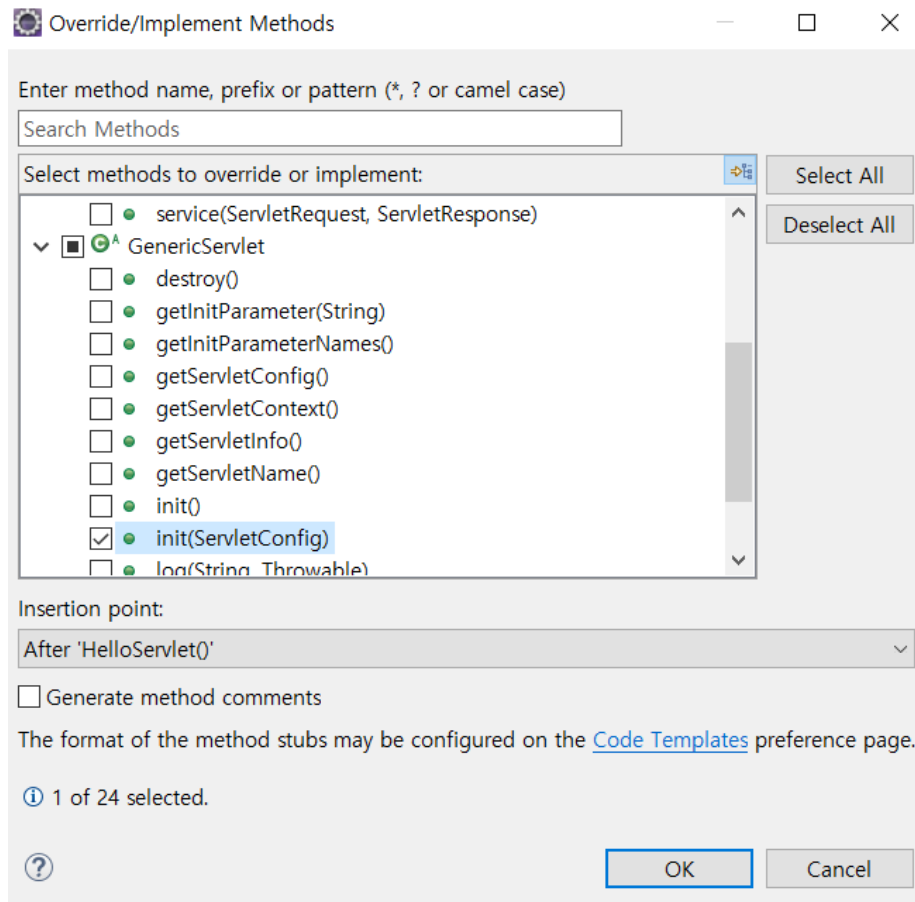


드래그한 부분을 지워주면 된다. (충돌이 나서 그랬을 경우가 큼)

Url mapping : /helloServlet 이랑 연결해주겠다



HelloServlet이라는 이름으로 들어오면 이 클래스한테 전달해줄게



- HelloServlet.java

```
package com.hello.controller;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletConfig;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 * Servlet implementation class HelloServlet
 */
@WebServlet("/HelloServlet")
public class HelloServlet extends HttpServlet {
    // 직렬화 (찾아보기)
    private static final long serialVersionUID = 1L;

    private String initParameter;
    private String contextParameter;

    public HelloServlet() {
        System.out.println("Hello Servlet 생성!!");
    }

    @Override
    public void init(ServletConfig config) throws ServletException {
        System.out.println("Hello Servlet init !!!");
        contextParameter = config.getServletContext().getInitParameter("url");
        System.out.println("contextParam : "+contextParameter);

        initParameter = config.getInitParameter("driver");
        System.out.println("initParam : "+initParameter);
    }
}
```

```

// get 방식으로 넘어온 요청을 처리
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
// 무조건 인코딩부터 하고 시작!!!!!! 진짜진짜 중요!!!!
request.setCharacterEncoding("UTF-8");
response.setContentType("text/html; charset=UTF-8");

System.out.println("GET 방식으로 요청됨!");
System.out.println("init Param : "+initParameter);

String command = request.getParameter("command");
System.out.println("<" + command + ">");

PrintWriter out = response.getWriter();
out.append("<h1 style='background-color:dodgerblue;'>Hello Servlet</h1>");
out.append("<h2>계층구조 / 라이프 사이클 / url 매핑</h2>");
out.append("<a href='home.html'>돌아가기</a>");

}

// post 방식으로 넘어온 요청을 처리
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
request.setCharacterEncoding("UTF-8");
response.setContentType("text/html; charset=UTF-8"); // 오타나면 다운로드 된다.

System.out.println("POST 방식으로 돌아옴!!");
System.out.println("initParam : " + initParameter);

String command = request.getParameter("command");
System.out.println("[" + command + "]");

response.getWriter().append("<h1 style='background-color=pink;'>Hello Servlet</h1>")
.append("<h2> 계층구조 / 라이프 사이클 / url 매핑</h2>")
.append("<a href='home.html'>돌아가기</a>");

}

@Override
public void destroy() {
System.out.println("servlet destroy!!!");
}
}

```

프로젝트 우클릭 → JAVA EE Tools → Generate 누르면 web.xml 생긴다.

```

> JavaScript Resources
> build
v WebContent
> META-INF
v WEB-INF
> lib
  web.xml
  home.html

```

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee" xsi:schemaLocation="http://xmlns.jcp.
3 <display-name>Jsp05_HelloServlet</display-name>
4 <welcome-file-list>
5 <welcome-file>home.html</welcome-file>
6 <welcome-file>index.htm</welcome-file>
7 <welcome-file>index.jsp</welcome-file>
8 <welcome-file>default.html</welcome-file>
9 <welcome-file>default.html</welcome-file>
10 <welcome-file>default.jsp</welcome-file>
11 </welcome-file-list>
12
13 <context-param>
14 <param-name>url</param-name>
15 <param-value>jdbc:oracle:thin:@localhost:1521:xe</param-value>
16 </context-param>
17
18 <servlet>
19 <servlet-name>hello</servlet-name>
20 <servlet-class>com.hello.controller.HelloServlet</servlet-class>
21 <init-param>
22 <param-name>driver</param-name>
23 <param-value>oracle.jdbc.driver.OracleDriver</param-value> <!-- 지금..값은 아무의미가 없다..adc 써놔도됨 -->
24 </init-param>
25 <load-on-startup>1</load-on-startup>
26 </servlet>
27
28 <servlet-mapping>
29 <servlet-name>hello</servlet-name>
30 <url-pattern>/hellocontroller.do</url-pattern>
31 </servlet-mapping>
32
33 </web-app>

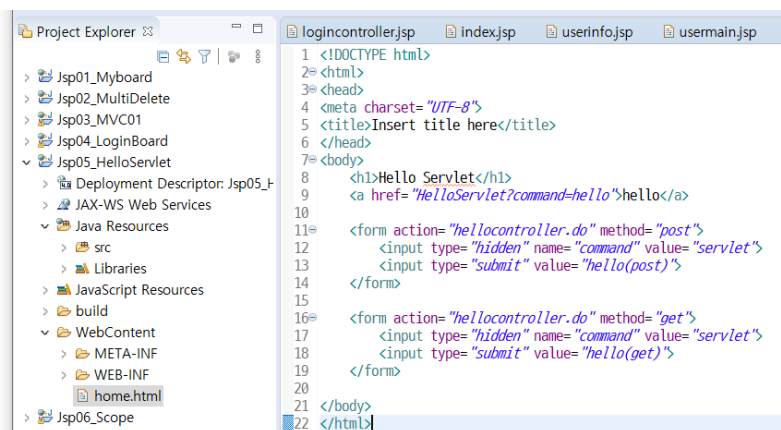
```

Hello Servlet

hello

hello(post)

hello(get)

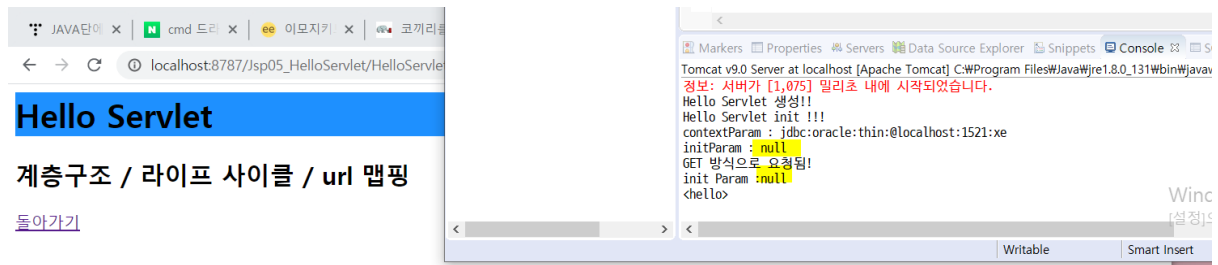


```

Markers Properties Servers Data Source Explorer Snippets Console
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre1.8.0_131\bin\Wj
정보: 적어도 하나의 JAR가 TLD들을 찾기 위해 스캔되었으나 아무 것도 찾지 못했습
Hello Servlet 생성!!
Hello Servlet init !!!
contextParam : jdbc:oracle:thin:@localhost:1521:xe
initParam : oracle.jdbc.driver.OracleDriver
7월 01, 2020 1:03:41 오전 org.apache.jasper.servlet.TldScanner scanJars
정보: 적어도 하나의 JAR가 TLD들을 찾기 위해 스캔되었으나 아무 것도 찾지 못했습
7월 01, 2020 1:03:41 오전 org.apache.coyote.AbstractProtocol start
정보: 프로토콜 핸들러 ["http-nio-8787"]을(를) 시작합니다.

```

hello 클릭시

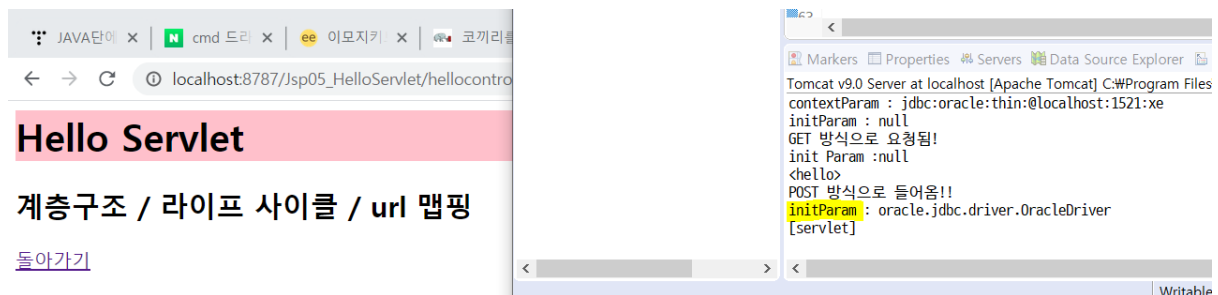


get방식으로 들어가게되면 web.xml 파일을 거치지 **않고** 가기 때문에 web.xml에서 정의해준

```
<init-param>
  <param-name>driver</param-name>
  <param-value>oracle.jdbc.driver.OracleDriver</param-value>
</init-param>
```

init-param의 값을 가져오지 못한다. 그래서 콘솔에는 null값이 찍힌다.

hello(post) 버튼 클릭시



post방식으로 들어가게되면 web.xml 파일을 거쳐서 가기 때문에 init-param의 값을 같이 갖고 간다.

그래서 콘솔에는 init의 값이 제대로 찍히게 된다.

hello(get) 버튼 클릭시

혹시 이게 get과 post 방식의 차이뿐인가? 대답만을 말하자면 그렇지 않다.

```
<form action="hellocontroller.do" method="get">
  <input type="hidden" name="command" value="servlet">
  <input type="submit" value="hello(get)">
</form>
```

어.디.로 보내느냐가 중요한 것 같다.

action 속성값에 servlet을 찾아서 실행시키라고 url-pattern(hellocontroller.do)를 줬더니

web.xml에 있는 값을 잘 찾아서 init-param 과 같이 전달이 되었다.

그래서 콘솔에 init-param의 값도 null이 아닌 가져온 값이 잘 찍혔다.

SCOPE

scope : 객체 전달 범위

1. page : 현재 페이지에서 객체를 전달 (현재 페이지의 객체 유지)
2. request : 현재 페이지에서 요청한 다음 페이지까지만 객체 전달 (요청된 페이지까지 객체 유지)
3. session : 현재 Browser 에서 context 내에 있는 모든 페이지에서 객체 유지

4. application : 현재 context 내에서 객체 유지. Web.xml 에서도 사용 가능

공통메소드 : setAttribute, getAttribute, removeAttribute, getAttribute

- ScopeController.java

```
package com.scope.controller;

import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

@WebServlet("/ScopeController")
public class ScopeController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        doPost(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        response.setContentType("text/html; charset=UTF-8");

        String requestId = (String) request.getAttribute("requestId");
        System.out.println("requestId: "+requestId);

        HttpSession session = request.getSession();
        String sessionId = (String) session.getAttribute("sessionId");
        System.out.println("sessionId : "+sessionId);

        ServletContext application = getServletContext();
        String applicationId = (String) application.getAttribute("applicationId");
        System.out.println("applicationId : "+applicationId);

        PrintWriter out = response.getWriter()
            .append("<h1>Controller에서 만들어진 table</h1>"
                + "<table border='1'>"
                + "<tr>"
                + "<th>requestId</th>"
                + "<td>"+requestId+"</td>"
                + "</tr>"
                + "<tr>"
                + "<th>sessionId</th>"
                + "<td>"+sessionId+"</td>"
                + "</tr>"
                + "<tr>"
                + "<th>applicationId</th>"
                + "<td>"+applicationId+"</td>"
                + "</tr>"
                + "</table>");
    }
}
```

- index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("UTF-8");
    response.setContentType("text/html; charset=UTF-8");
%>
```

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
    pageContext.setAttribute("pageId", "my pageId value!!");
    request.setAttribute("requestId", "my requestId value!!");
    session.setAttribute("sessionId", "my sessionId value!!"); // session은 invalidate전까지
    application.setAttribute("applicationId", "my applicationId value!!");
%>
    <h1>index</h1>

    pageId : <%=pageContext.getAttribute("pageId") %> <br/>
    requestId : <%=request.getAttribute("requestId") %> <br/>
    sessionId : <%=session.getAttribute("sessionId") %><br/>
    applicationId : <%=application.getAttribute("applicationId") %><br/>

    <a href="result.jsp?param=myrequestparamvalue">result...</a>
    <br>
    <a href="controller.do">controller🍷</a>
    <% pageContext.forward("controller.do"); %>
</body>
</html>

```

- result.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    request.setCharacterEncoding("UTF-8");
    response.setContentType("text/html; charset=UTF-8");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>

    <h1>result🍷</h1>

    pageId : <%=pageContext.getAttribute("pageId") %> <br/>
    requestId : <%=request.getAttribute("requestId") %> <br/>
    sessionId : <%=session.getAttribute("sessionId") %><br/>
    applicationId : <%=application.getAttribute("applicationId") %><br/>

    requestParam : <%=request.getParameter("param") %> <br/>

</body>
</html>

```

index페이지에 마지막 pageContext.forward("controller.do") 를 주석처리 하고 실행시키면

← → ↻ localhost:8787/Jsp06_Scope/

index

pageld : my pageld value!!
requestId : my requestId value!!
sessionId : my sessionId value!!
applicationId : my applicationId value!!
[result...](#)
[controller](#) 🍓

모든 값이 유지가 잘 되는 것을 볼 수 있다.

result 를 클릭했을때 과연 모든 값들이 다 유지가 될까?

처음 생각으로선 pageld : null 을 제외한 나머지는 다 값이 있을 거라고 생각했다. 하지만

← → ↻ localhost:8787/Jsp06_Scope/result.jsp?param=myrequestparamvalue

result 🐻

pageld : null
requestId : null
sessionId : my sessionId value!!
applicationId : my applicationId value!!
requestParam : myrequestparamvalue

param값은 location.href 뒷부분 url에 name : value 로 값을 넘겨줬기 때문에 잘 나오고

session, application 은 페이지가 열마가 이동이 되든 값(객체)를 가지고 다니기 때문에 잘 나오고

requestId는 왜 안나올까?

request는 요청한 페이지의 다음 페이지까지만 객체가 유지가 되는데

index.jsp → web.xml → result.jsp 로 가기 때문에 객체가 휘발이 되어 버린다.

마찬가지로 index.jsp로 돌아와서 controller를 클릭해보면

← → ↻ localhost:8787/Jsp06_Scope/controller.do

Controller에서 만들어준 table

requestId	null
sessionId	my sessionId value!!
applicationId	my sessionId value!!

index.jsp → web.xml → ScopController.java 로 와서 PrintWriter로 response를 해주는데 테이블을 같이 만들어 값을 보여준다. requestId이 null 인것을 볼 수가 있다.

그렇다면 requestId가 나오게 request의 객체를 어떻게 유지시킬까? → forward를 써야한다.

index.jsp에서 주석처리해준 pageContext.forward("controller.do") 를 주석해제 하고 다시 실행

← → ↻ localhost:8787/Jsp06_Scope/

Controller에서 만들어준 table

requestId	my requestId value!!
sessionId	my sessionId value!!
applicationId	my sessionId value!!

index.jsp → web.xml → ScopController.java

많은 페이지 이동이 있어도 request를 포함한 모든 객체가 유지가 된다. forward만세

(같은 servlet을 사용한다고 해도 href 처럼 바로 url-pattern을 사용해 보내주느냐

forward에 url-pattern을 사용해 보내주느냐의 차이는

주소창에 보내준 곳의 이름이 나오느냐.안나오느냐의 차이다)