

[dev-rary(데브러리)] JSP 빠른 스캔(원리X)

1. form과 request

前) 광고데이터 분석, SEO 최적화, SEM 세팅 및 관리

前) 기초프로그래밍, 웹개발, 앱개발 강의

前) 머신러닝을 활용한 데이터 분석

前) 기아자동차 광주공장 외주

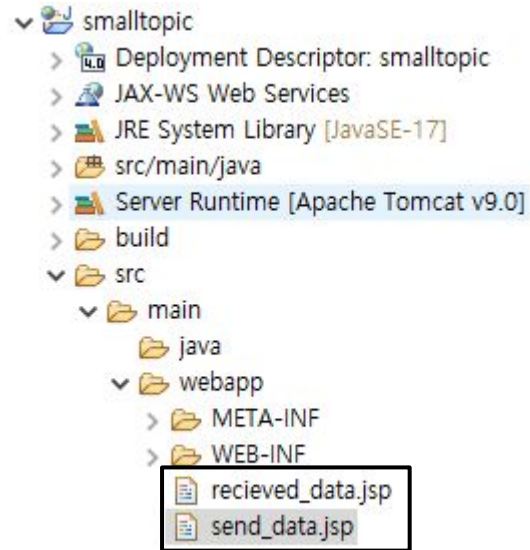
前) 리그오브 레전드 데이터 분석 등...

現) 국비반 강의 진행중

現) 유튜브 데브러리의 개발도서관 운영중

<https://www.youtube.com/channel/UCJJsnLh62qx00sN3yzRcW3Q>

기초 자바에서는 눈에 보이는 코드만 신경써도 되었는데
점점 여러 클래스파일을 넘나들며 호출 순서를 고려해야 하고
웹개발에서는 보이지 않는 데이터의 전송이 많아 어려우셨을 겁니다.
본 토막강의 영상에서는 정말 간단하게 메모리상의 변화만 다룹니다.
원리를 다루지 않기때문에 이 동영상만으로는 절대 100% 현업에서 요구하는
지식을 충족시킬수 없으니 심화된 내용을 공부하기 위한 토대로만 삼아주세요



send_data.jsp(송신 데이터)

파일에 form을 만들어 데이터를 넘겨주는 부분을 먼저 살펴보고

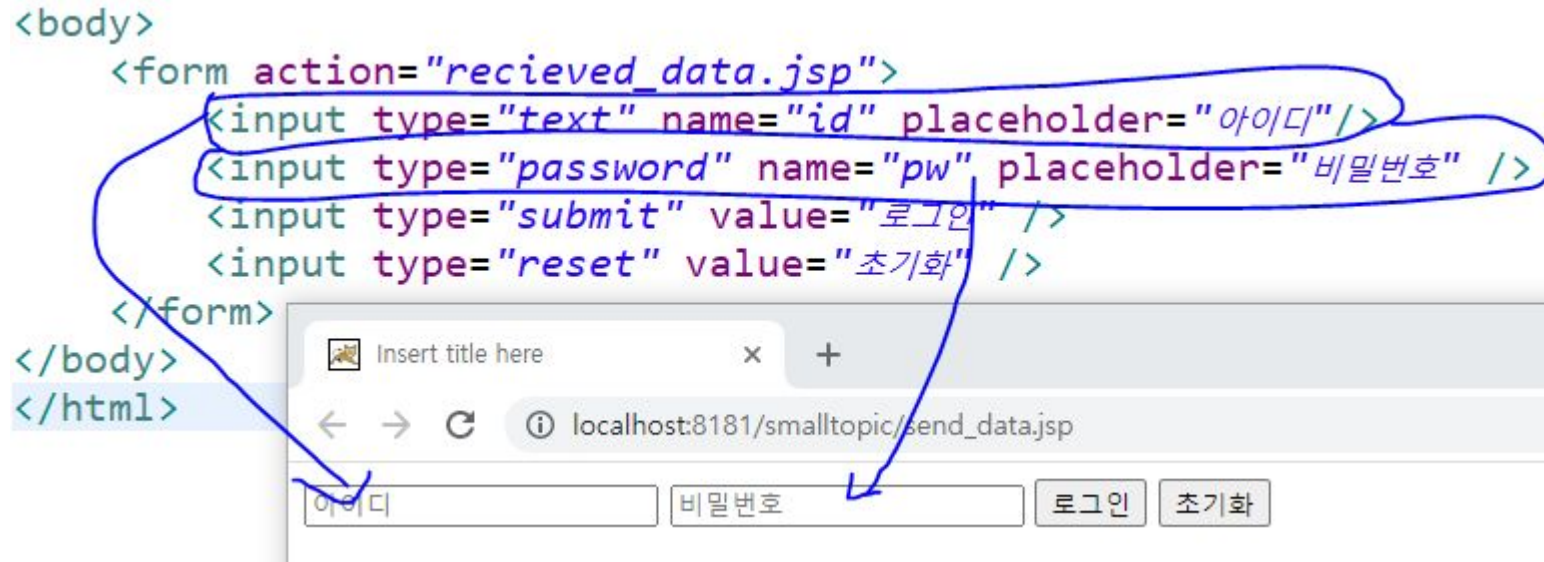
recieved_data.jsp(수신 데이터)

파일에 send_data.jsp에서 보낸 데이터가 자바 데이터화 되는 과정을 봅니다.

먼저 send_data.jsp에 form태그를 작성했습니다.

1번의 action="목적지" 로 데이터가 전달되며

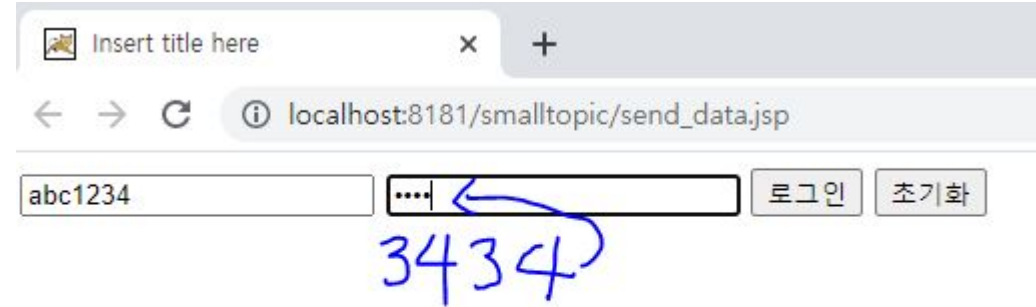
2번, 3번의 name에 적힌 이름에 입력된 데이터를 저장해 전달합니다.



위와 같이 데이터를 담을 수 있는 폼이 완성됩니다.

이제 아이디와 비밀번호를 기입하면 생기는 일에 대해 보겠습니다.

```
<form action="recieved_data.jsp">
  <input type="text" name="id" placeho
  <input type="password" name="pw" pla
  <input type="submit" value="로그인" />
  <input type="reset" value="초기화" />
</form>
```

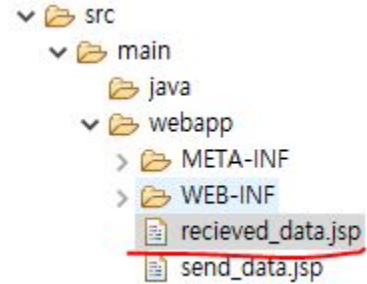


목적지 : recieved_data.jsp
id변수에 저장된 값 : abc1234
pw변수에 저장된 값 : 3434

위 코드의 관계는, 좌측 name에 적힌 “id”, “pw”를 변수명으로 생각하시고

우측 abc1234, 3434는 id, pw라는 변수에 각각 저장될 자료입니다.

로그인시 아래 표처럼 포장해 recieved_data.jsp라는 목적지로 배송합니다.



목적지 : recieved_data.jsp

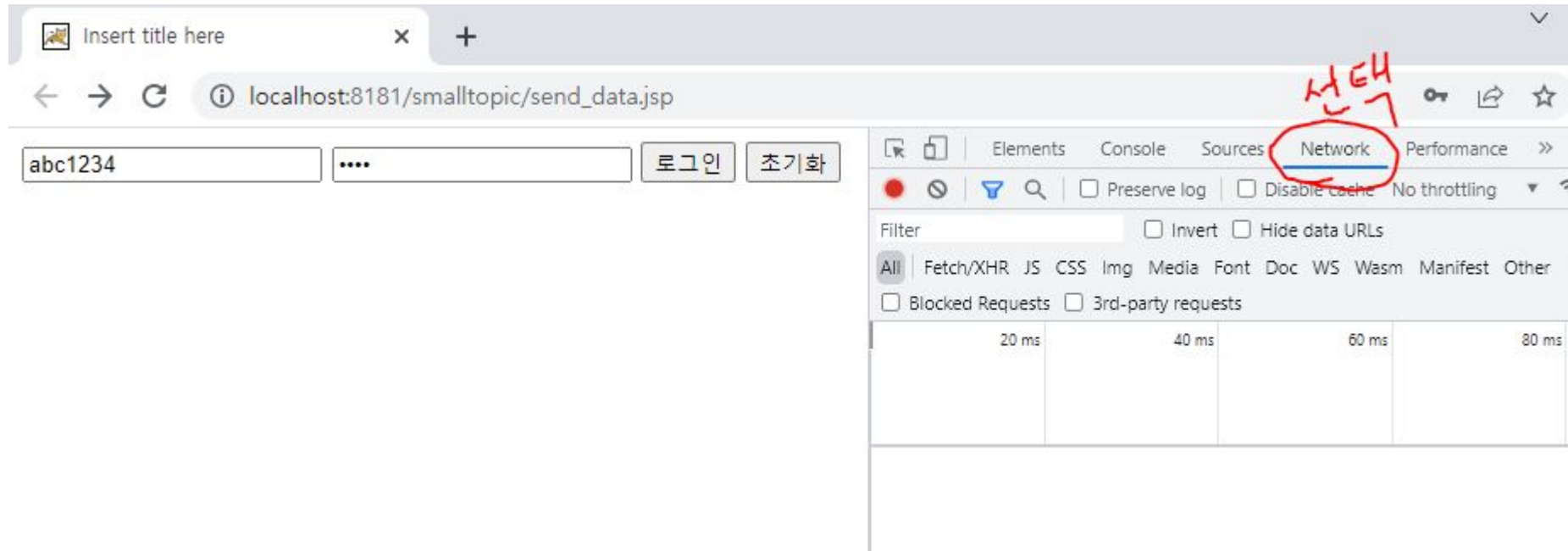
id변수에 저장된 값 : abc1234

pw변수에 저장된 값 : 3434

발송한 데이터는 위 그림처럼 recieved_data로 전송되게 되며,

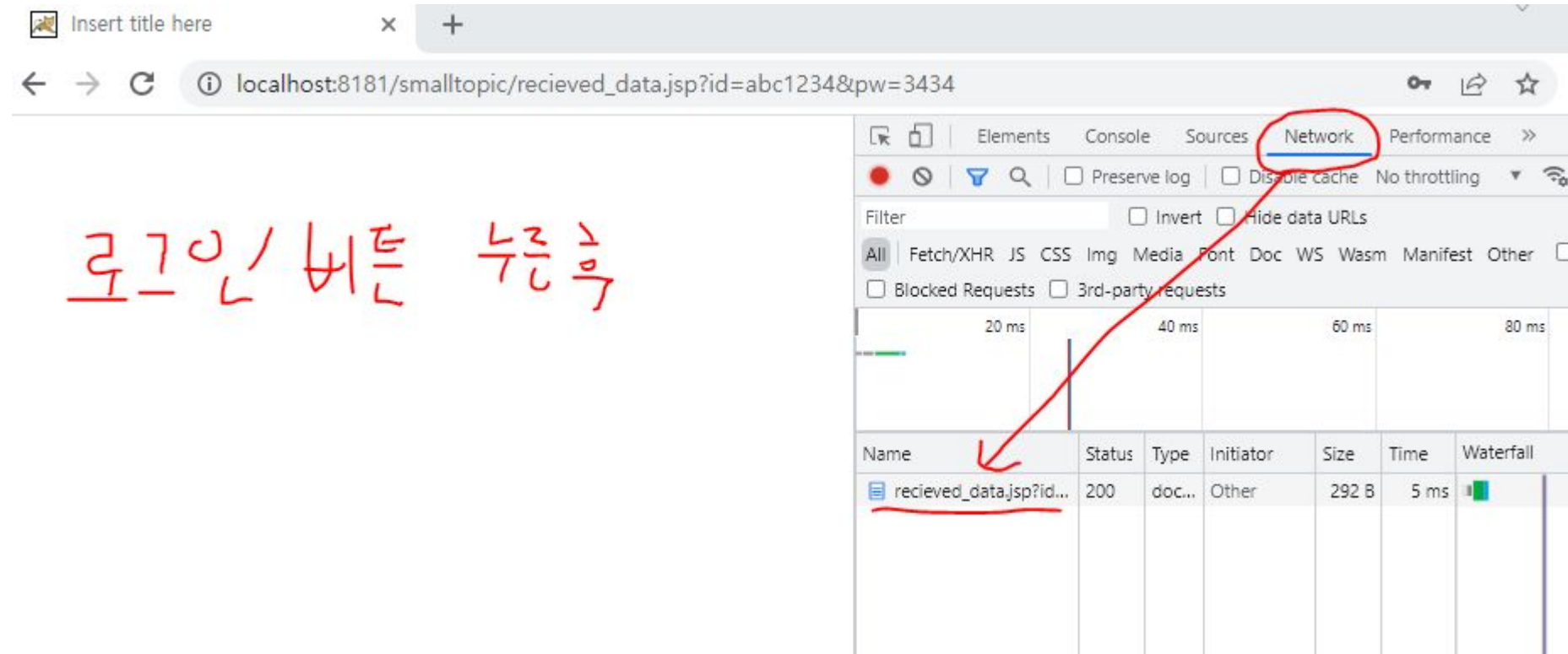
id라는 http 변수에 abc1234가

pw라는 http 변수에 3434가 담겨서 저장됩니다.

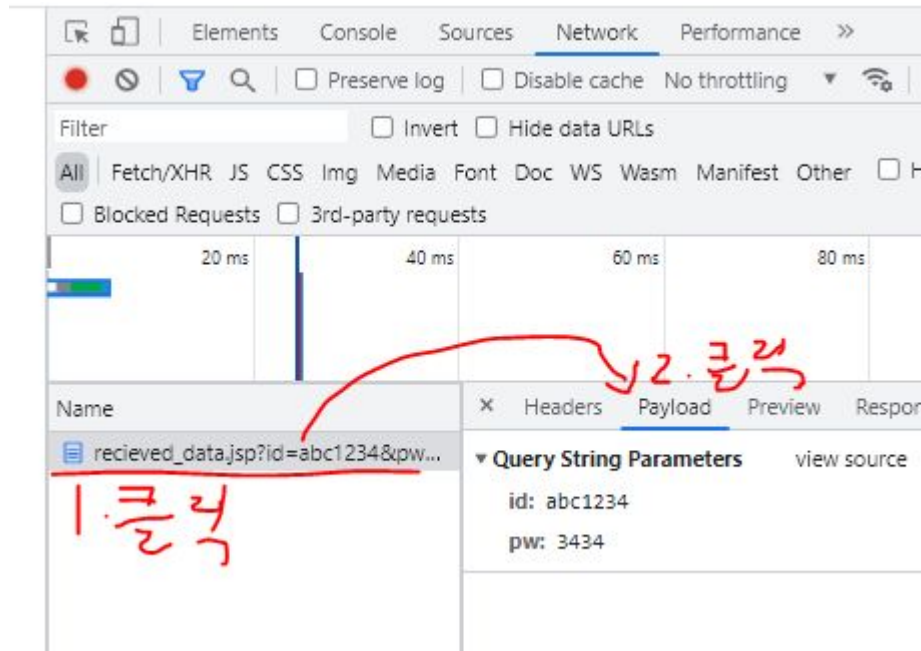


이를 확인하는 방법은 `send_data.jsp`에서 로그인 버튼을 누르기 전에 `f12`를 눌러서 먼저 개발자 도구 창을 켜고, **Network** 탭을 들어가줍니다.

recieved_data.jsp 메모리 구조를 통해 수신데이터 확인하기

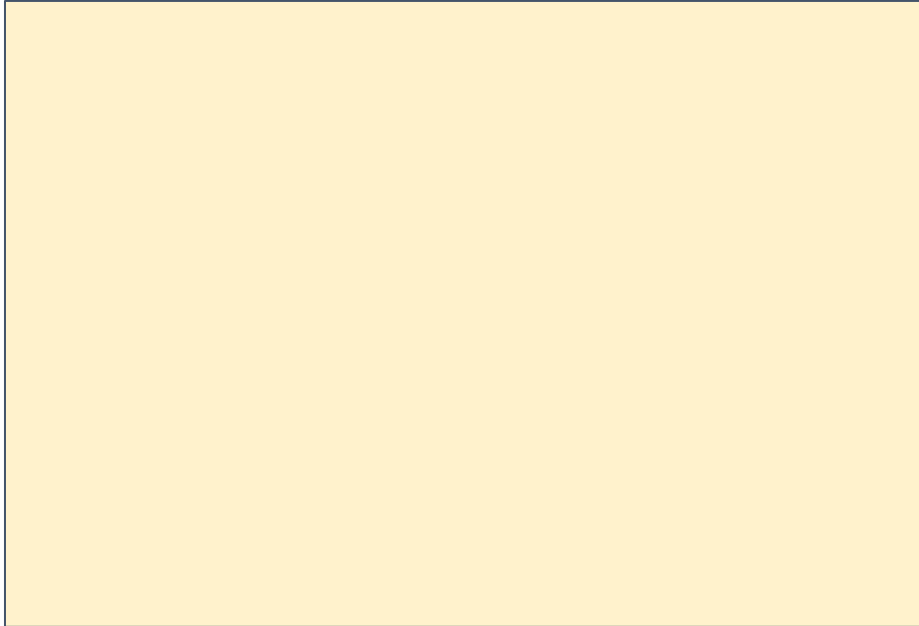


로그인버튼을 눌러서 넘어오면 하단 탭에 목적인 **recieved_data.jsp** 항목이 추가됩니다. 이제 저 항목을 클릭해보겠습니다.

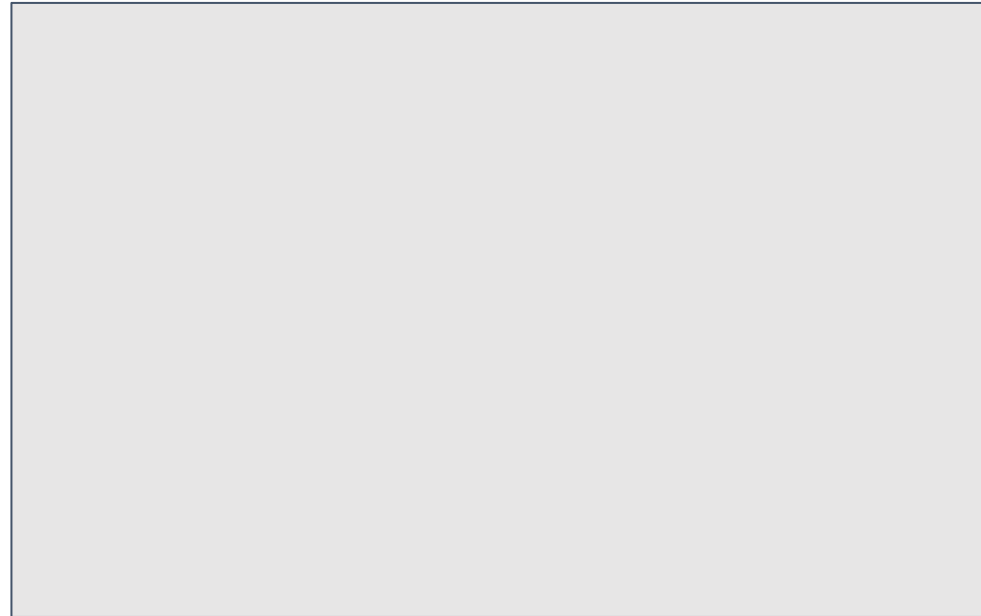


name탭에서 recieved_data.jsp를 클릭한 후 Payload 탭을 클릭해보면
위와 같이 id에는 abc1234가 담겨서, pw에는 3434가 담겨서 전송되는것을
볼 수 있습니다.

휘발성 공간(페이지 이동시 삭제)



비휘발성 메모리 공간(페이지 이동에도 데이터 유지)

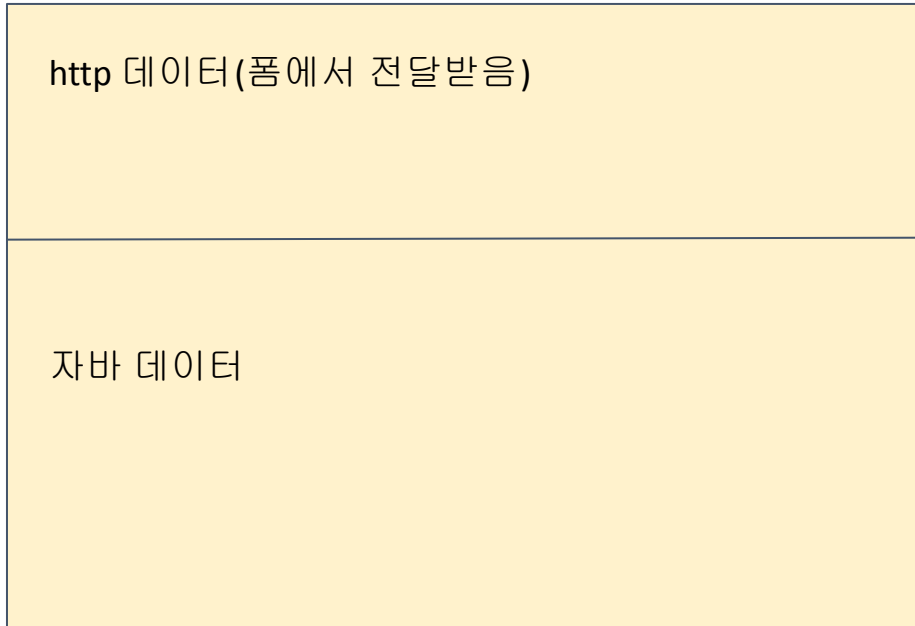


이제 **JSP**의 메모리 구조에 대해서 보겠습니다.

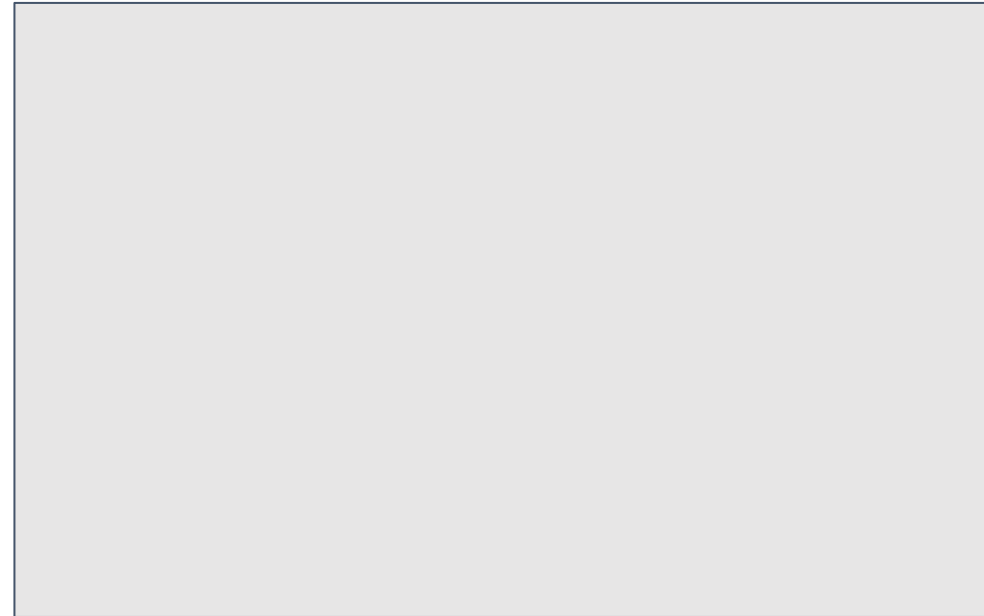
휘발성(일시적)메모리 공간과 비휘발성(지속적) 메모리 공간이 있습니다.

공식 용어는 아니고 이해를 돕기 위해 제가 설명하는 방식입니다.

휘발성 공간(페이지 이동시 삭제)

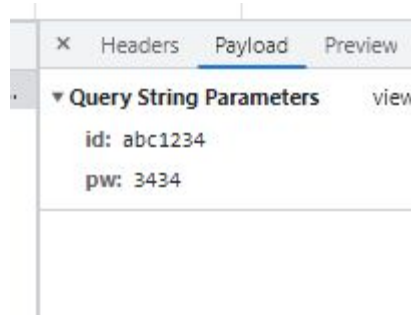


비휘발성 메모리 공간(페이지 이동에도 데이터 유지)



페이지 이동시 삭제되는 휘발성 메모리 공간(recieved_data.jsp)은 크게
자바 데이터(내부적으로 변수를 선언해 데이터를 저장) 공간과
http데이터(form 태그를 통해 전달된 데이터를 저장) 공간으로 나뉩니다.

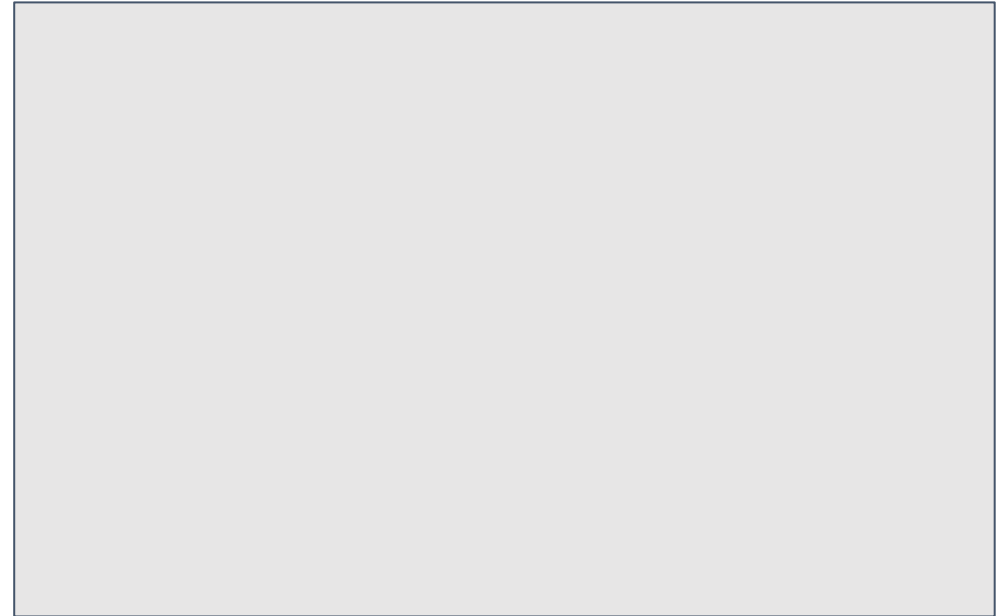
recieved_data.jsp공간(페이지 이동시 삭제)



http 데이터(폼에서 전달받음)
id : abc1234
pw : 3434

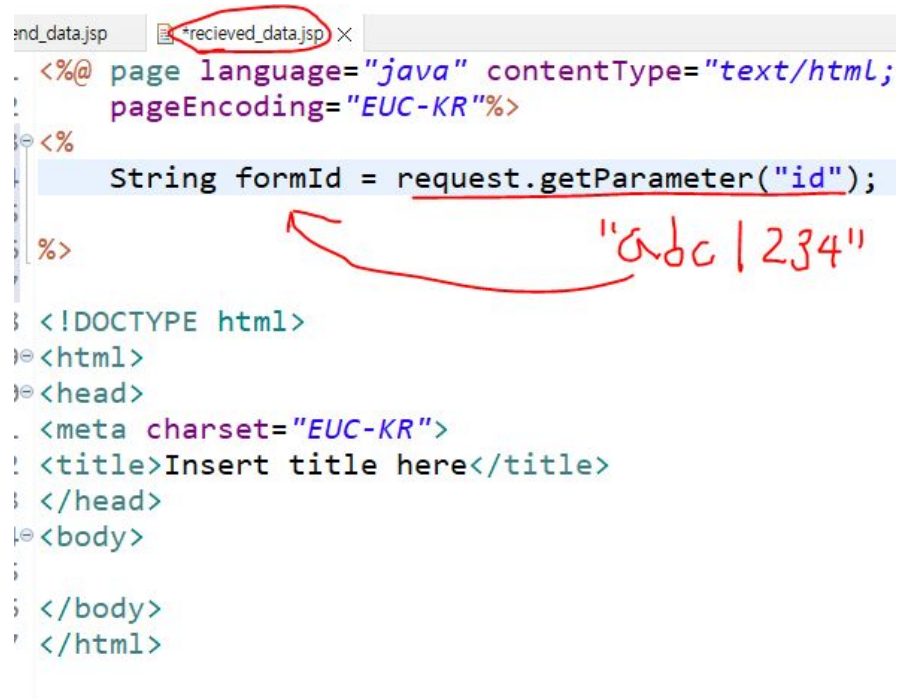
자바 데이터

비휘발성 메모리 공간(페이지 이동에도 데이터 유지)



http데이터에는 아까 폼에서 전달받은 데이터인 id, pw가 저장됩니다.

아직 자바 내부에서는 쓸 수 없고, 단지 http 데이터로서만 전달된 상태입니다.



```
recieved_data.jsp
<%@ page language="java" contentType="text/html;
pageEncoding="EUC-KR"%>
<%
String formId = request.getParameter("id");
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="EUC-KR">
<title>Insert title here</title>
</head>
<body>
</body>
</html>
```

recieved_data.jsp공간(페이지 이동시 삭제)

http 데이터(폼에서 전달받음)
id : abc1234
pw : 3434

자바 데이터
formId : abc1234

위에서는 `request.getParameter("id");` 를 호출했으니 id에 매칭된
abc1234를 가져오게 되고, 자바데이터로 바뀐 해당 자료를
자바 내 변수인 formId에 저장합니다.

recieved_data.jsp공간(페이지 이동시 삭제)

```
send_data.jsp  recieved_data.jsp x
1  <%@ page language="java" contentType="text/html; ch
2  pageEncoding="EUC-KR"%>
3  <%
4      String formId = request.getParameter("id");
5      String formPw = request.getParameter("pw");
6  %>
7
8  <!DOCTYPE html>
9  <html>
10 <head>
11 <meta charset="EUC-KR">
12 <title>Insert title here</title>
13 </head>
14 <body>
15
16 </body>
```

http 데이터(폼에서 전달받음)

id : abc1234

pw : 3434

자바 데이터

formId : abc1234

formPw : 3434

마지막으로 pw도 받아서 자바 데이터로 바꾸면

이제 http데이터 영역에서 자바 데이터 영역으로 데이터가 다 옮겨져

스크립트릿 내부적으로 쓸 수 있는 자료가 됩니다.

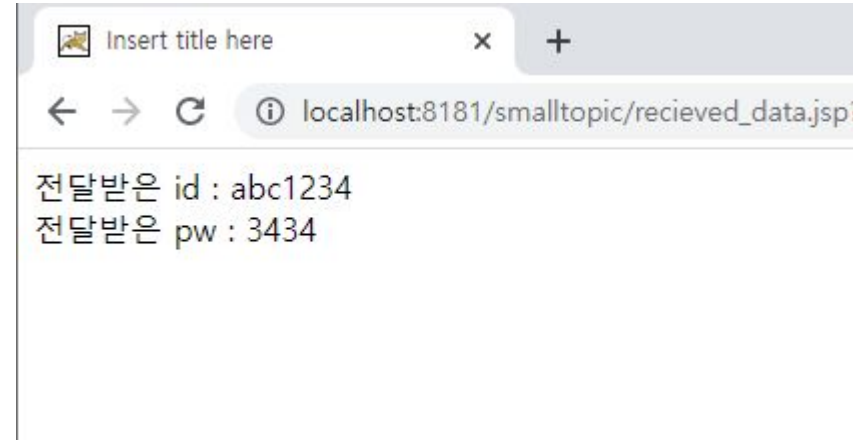

```
1 <%@ page language="java" contentType="text/html; ch
2   pageEncoding="EUC-KR"%>
3 <%
4   String formId = request.getParameter("id");
5   String formPw = request.getParameter("pw");
6   System.out.println("전달받은 id : " + formId);
7   System.out.println("전달받은 pw : " + formPw);
8   out.write("전달받은 id : " + formId + "<br/>");
9   out.write("전달받은 pw : " + formPw + "<br/>");
10 %>
11
12 </DOCTYPE html>
```

이제 해당 데이터가 들어왔음을 확인하기 위해

`System.out.println()`을 이용해 콘솔에 찍어서 확인하거나

`out.write()`나 `out.print()` 등의 화면에 찍어서 확인하면 됩니다.

```
Tomcat v9.0 Server at localhost [Apache Tomcat] C:\eclipse\workspace-jee-2021-  
2월 19, 2022 7:07:09 오후 org.apache.c  
정보: 웹 애플리케이션 디렉토리 [C:\apache-to  
2월 19, 2022 7:07:09 오후 org.apache.c  
정보: 프로토콜 핸들러 ["http-nio-8181"]을(  
2월 19, 2022 7:07:09 오후 org.apache.c  
정보: 서버가 [1227] 밀리초 내에 시작되었습니다.  
전달받은 id : abc1234  
전달받은 pw : 3434
```



콘솔과 화면 결과입니다.

send_data.jsp 에서 여러분들이 다른 문자를 기입해서도 전달해보시고

post 방식으로도 해 보시고, 폼 추가도 해서 확실히 이해하고 넘어가주세요.