

클라잍언트가 폼에 입력하는 거 어떻게 꺼내서 활용하냐

2025년 상반기 K-디지털 트레이닝

스코프중에서 젤 많이 쓰이는 2가지 리퀘스트 스코프 세션 스코프

HTML5 Form 태그와 서블릿

[KB] IT's Your Life

쿠키를 사용해서 정보를 더 오래 사용하는 법

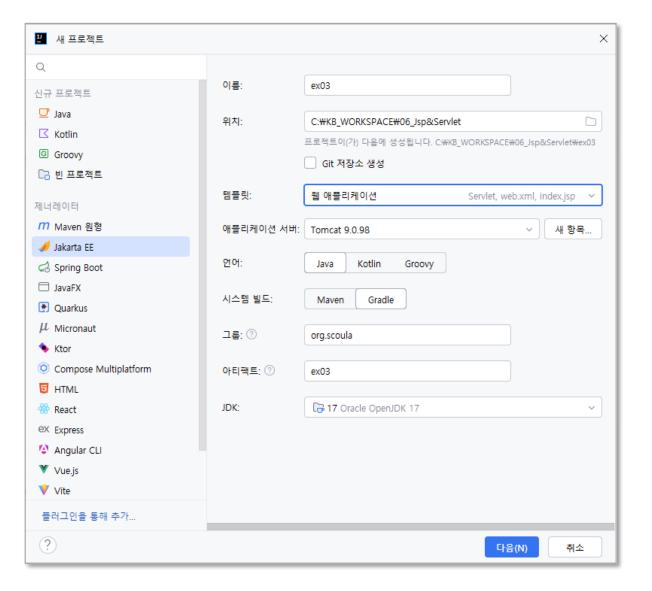


프로젝트 만들기

💟 프로젝트 생성

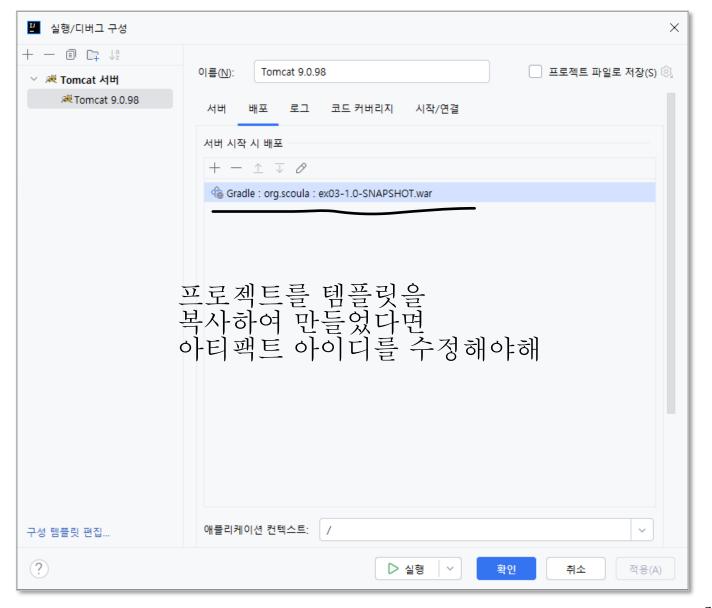
- o Project name: ex03
- o Version: Java EE 8

새프로젝트 만들고 자카르타로



Tomcat 설정

o context path: /



HTML 폼 태그

폼 필드의 컴포넌트	사용방법	
홍길동	(input type="text" name="username" value="홍길동")	
	(input type="password" name="passwd" value="1234")	
 찿아보기	(input type="file" name="theFile")	
□value1 ☑value2 ☑value3	<pre>(input type="checkbox" name="check" value="value1") (input type="checkbox" name="check" value="value2") (input type="checkbox" name="check" value="value3")</pre>	
⊕ value1 • value2 ⊕ value3	<pre>(input type="radio" name="check" value="value1") (input type="radio" name="check" value="value2") (input type="radio" name="check" value="value3")</pre>	
[일반단추]	(input type="button" name="button" value="일반단추")	
멀티라인 ^ (a)	<pre></pre>	
보내기단추	(input type="submit" value="보내기단추")	
지우기단추	(input type="reset" value="지우기단추")	

2 HTML5의 Form

🧿 form 태그

- o action 속성
 - submit 버튼 선택시, 웹 서버에서 처리할 커포넌트 지정
 - Jsp나 servlet 경루 지정
- o method 속성
 - 웹 브라우저에서 웹 서버로 요청하는 방법 명시
- o GET 방식 V
 - 요청 파라미터 값이 웹 브라우저의 URL에 명시되어 웹 서버로 전송 http://서버 IP번호:포트번호/컨텍스트명/경로명/login?name=홍길동&age=20
 - <u>서블릿에서 doGet 메서드가 처리</u> J√V ✓ ✓
 - 파리미터 길이에 제한
 - <u>보안에 취약</u>
 - 일반 태그의 링크 및 명시적 URL 요청

2 HTML5의 Form

o POST 방식 √

- 요청 파라미터 값이 HTTP Request의 요청 몸체(body)에 포함되어 전송 (URL에서 확인 불가) http://서버 IP번호:포트번호/컨텍스트명/경로명/login
- <u>서블릿에서 doPost() 메서드에서 처리</u>
- 새로 고침하는 경우 사용자에게 재요청 의사 묻는 정보창을 실행

서블릿에서는 post나 get이나 상관없음.

```
<form action="/login">// method 생략시 기본값 get<form action="/login" method="get">// get 방식으로 전송<form action="/login" method="post">// post 방식으로 전송
문자셋정도만
```

정보를 빼내는 방법은 같아

♡ 파라미터 값 추출 : request 객체 메서드 이용

어덯게 꺼낼까

리턴 타입	메서드명 꺼낼때 메소드	내용
String	getParameter(name) 폼태그에서 지정했던	name에 해당하는 파라미터 값을 리턴. 타입에 따라 문자열이 지정한 name의 파라미터 값이 없으면 null 리턴. 비어있을수도 잇어
String[]	name속성의 밸류 getParameterValues(name) 값들을 배열로	checkbox, select 태그와 같이 하나의 name에 여러 값을 가지는 경우에 사용. name에 해당되는 파라미터 값을 배열로 리턴.
Enumeration	다중선택한 경우 getParameterNames()	<u>폼 태그 내에 파라미턴</u> name 목록을 얻고 <u>자하는 경우 사용.</u> 모든 name 값을 Enumeration 타입으로 리턴.

키-밸류쌍으로 데이터가 되어있겠지

리퀘스트 객체에 캡슐화되어 정보들이 들어있다.

그 처리는 서블릿 앞단에서 끝나있음.

즉 서블릿은 같은 리퀘스트 객체만 이용하면 돼. 활용하여 처리하자

신경써야하는 것중 하나 한글처리

- ✓ 서블릿의 한글 처리 get post 차이 있음
 - lo GET 방식 표준이 UTF-8 로 url이 디폴트임 표준임
 - URL을 UTF-8로 인식하기 때문에 한글 인코딩 지정 필요없음
 - o POST 방식 해당 운영체제의 디폴트 문자셋으로 처리하기 떄문에
 - 톰캣의 디폴트 문자 인코딩 방식: ISO-8859-1 윈도우에서 디폴트 문자 셋
 - ▶ HTTP BODY에 있는 입력 파라미터 값에 한글이 포함되어 있는 경우 한글 인코딩 작업 필요 전송할때 인코딩과 받아서 해석하는 이코딩과 차이가 난다. request.setCharacterEncoding("UTF-8");

받을때도 utf-8로 지정해야함. 받기전에

3

🦁 getParameter(name) 메서드

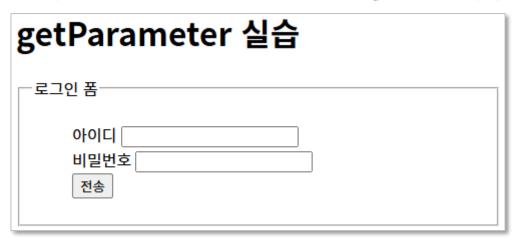
- 하나의 파라미터 값을 추출
- \circ name에 해당하는 파라미터가 없다면 \circ null 리턴 (오타 주의) \checkmark \lor

☑ login_form.jsp 로그인 폼에서 로그인 ㄱㄱ하게

```
<body>
<h1>getParameter 실습</h1>
                                                   액ㅂ션속성값 기억. 경로가 상대경로임.
<<u>form action="login" method="get"></u>
   <fieldset>
      <legend>로그인 폼</legend>
      <
            <label for="userid">0|0|C|</label>
            <input type="text" id="userid" name="userid">
                                                       id는 톰캣에 안넘어가
태그 내에서 id속성은 JS에서 제어할때
또는 라벨태그와 연결될때.
         <
            <label for="passwd">비밀번호</label>
                                                       서버와는 id속성은 상관 없다
            <input type="password" id="passwd" name="passwd">
         <input type="submit" value="전송">
                                                       서버와는 name속성이 연관이 잇는것.
      </fieldset>
</form>
</body>
```

Jsp 확인

http://localhost:8080/ex03/login_form.jsp



서블릿 만들기

- 패키지: org.scoula.ex03
- 서블릿명: LoginServlet
- o URL 맵핑명: /login

주의 url매핑명을

연결할 폼태그의 action속성값이랑 동일하게 맞춰야 함.

LoginServlet.jsp

```
어노테이션에서 매핑해주면 돼. 매핑된 폼태그는 겟 메소드를 사용. 서블릿에서는 doGet이 @WebServlet("/login") 호출됨. doGet을 오버라이딩해야해 public class LoginServlet extends HttpServlet {
 @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
       String userid = req.getParameter("userid"); 인풋 태그의 name속성값을 넣어줘야함.
String passwd = req.getParameter("passwd"); 항상 문자열을 반환함. html은 텍스트 기반이잖아?
        // 응답 내보내기
        resp.setContentType("text/html; charset=UTF-8");
                                                                              텍스트로 받고 형변화하여 사용.
        PrintWriter out = resp.getWriter();
                                                                              입력값이 없을때는
반환값은
        out.print("<html><body>");
        out.print("아이디값: " + userid +"<br>");
                                                    서블릿에서 이거 만드는것 힘들었죠
        out.print("비밀번호값: " + passwd +"<br>");
        out.print("</body></html>");
                                                    JSP사용하는 이유
                                                    서블릿과 JSP연동은 다음 강의부터
```

☑ login_form.jsp (POST 요청 보내기)

```
<body>
<h1>getParameter 실습</h1>
                            이번엔 포스트 메서드로 넘길때
<form action="login" method="post">
   <fieldset>
                                           doPost호출되겠지
      <legend>로그인 폼</legend>
      <
            <label for="userid">아이디</label>
            <input type="text" id="userid" name="userid">
         <
            <label for="passwd">비밀번호</label>
            <input type="password" id="passwd" name="passwd">
         <input type="submit" value="전송">
      </fieldset>
</form>
</body>
```

LoginServlet.java (^O doPost Override)

```
      @WebServlet("/login")

      public class LoginServlet extends HttpServlet {

      protected void doGet(HttpServletRequest req, HttpServletResponse response) throws ServletException, IOException {

      // req.setCharacterEncoding("UTF-8"); // 한글 깨짐 방지 - 있을 때와 없을 때 결과 확인!!!

      doGet(req, resp);

      }

      F겠에다가 요청
```

디폴트(UTF-8인코딩 안 한 경우)

아이디값: íロロê¸,ëロロ 비밀번호값: 1234

UTF-8인코딩한 경우(req.setCharacterEncoding("UTF-8"))

아이디값: 홍길동 비밀번호값: 1234

3

- getParameterValues(name) 메서드
 - o checkbox, select 처럼 하나 이상의 값을 가지는 파라미터 추출 시
 - 동일한 파라미터 명으로 여러 개 전송한 경우 이를 추출할 때 사용
 - 값이 여러 개 이므로 배열(String [])처리 필요함

```
?sports=약구&sports=농구&sex=여자
```

String[] sports = request.getParameterValues("sports");

sports.jsp

```
<body>
                                                                 -좋아하는 운동 및 성별:
<h1>getParameterValues 실습</h1>
<form action="sports" method="post">
   <fieldset>
                                                                    남 💿 여 🔾
       <legend>좋아하는 운동 및 성별</legend>
                                                                     전송
       <
              <label for="baseball">야구</label>
              <input type="checkbox" id="baseball" name="sports" value="야구" >
              <label for="football">축구</label>
              <input type="checkbox" id="football" name="sports" value="축구">
              <label for="basketball">농구</label>
              <input type="checkbox" id="basketball" name="sports" value="농구">
          <
              <label for="male">남</label>
              <input type="radio" id="male" name="sex" value="남자" checked>
              <label for="female">여</label>
              <input type="radio" id="female" name="sex" value="여자">
          <input type="submit" value="전송">
       </fieldset>
</form>
</body>
```

http://localhost:8080/sports.jsp

```
getParameterValues 실습

- 좋아하는 운동 및 성별
- 야구 □ 축구 □ 농구 □
남 ② 여 ○
전송
```

이름이 같아여

서블릿 추가

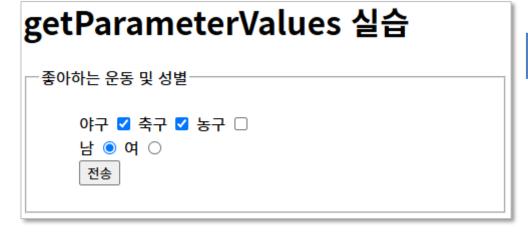
- o 패키지: org.scoula.ex03
- 서블릿명: SportsServlet
- o 맵핑명: /sports
- o 메서드: doPost

SportsServlet.java

```
@WebServlet("/sports")
public class SportsServlet extends HttpServlet {
 protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
   String[] sports = req.getParameterValues("sports");
   String sex = req.getParameter("sex");
   resp.setContentType("text/html; charset=UTF-8");
   PrintWriter out = resp.getWriter();
   out.print("<html><body>");
                                                  데이터들이 분리되어있음
   for (String sport : sports) {
     out.print("좋아하는 운동: " + sport + "<br>");
   out.print("성별 : " + sex + "<br>");
   out.print("</body></html>");
                                                      좋아하는 운동: 야구
                                                      좋아하는 운동: 축구
                                                       성별 : 남자
```

♡ 확인

http://localhost:8080/sports.jsp



→ http://localhost:8080/sports



좋아하는 운동: 야구 좋아하는 운동: 축구

성별 : 남자

board_form.jsp

```
<body>
  <h1>getParameterNames 실습</h1>
  <form action="board" method="post">
  <input type="hidden" name="action" value="write">
   제목: <input type="text" name="title"><br />
   작성자: <input type="text" name="author"><br />
   내용: <textarea name="content" rows="10" ></textarea><br />
   <input type="submit" value="저장">
                                                      http://localhost:8080/board_form.jsp
  </form>
                                                      getParameterNames 실습
</body>
                                                      제목: 테스트
                                                      작성자: 홍길동
                                                           내용입니다.
                                                      내용:
                                                       저장
```

🗸 서블릿 추가

- o 패키지: org.scoula.ex03
- 서블릿명: BoardServlet
- o 맵핑명: /board
- o 메서드: doGet, doPost

BoardServlet.java

```
@WebServlet("/board")
public class BoardServlet extends HttpServlet {
    protected void doGet(HttpServletRequest reg, HttpServletResponse resp) throws ServletException, IOException {
        resp.setContentType("text/html; charset=UTF-8");
        PrintWriter out = resp.getWriter();
        out.print("<html><body>");
                                                             요청에서 사용된 name속성값들
        Enumeration<String> enu = req.getParameterNames();
                                                                                    http://localhost:8080/board
        while(enu.hasMoreElements()) {
           String name = enu.nextElement(); \
String value = req.getParameter(name);
                                                                                    action: write
                                                                                    title : 테스트
            out.print(name + " : " + value + "⟨br⟩"); 		✓
                                                                                    author : 홍길동
                                                                                    content : 내용입니다.
        out.print("</body></html>");
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        req.setCharacterEncoding("UTF-8");
        doGet(reg, resp);
                                                                            모델데이터 객체로 캡슐화 ㄱㄱ
직접하게 되면 신경쓸게 아주많아요.
정보가 없을때. 타입변환. 예외처리.
데이터검증. 디폴트값 등등ㄷ등.
다행히 스프링같은 프레임웤에서 자동화
          정보들이 흩어져 잇는것이 아쉽다->캡슐화하자.
```