



It's Your Life

with





JSP의 기본 문법들



Scriptlet



Script

let

パイナップーパーン!
(Pineapple pen!)

사전

정의 출처: [Oxford Languages](#) · [자세히 알아보기](#)

스크립트, script

명사

1. 영화나 방송의 대본(臺本)·각본(脚本).
2. 인쇄
필기체의 활자.

스크립트 단어를 다음 언어로 번역:

언어 선택 ▼



무대

HTML







JS

JS

JS

사전

정의 출처: [Oxford Languages](#) · 자세히 알아보기



스크립트, script

명사

1. 영화나 방송의 대본(臺本)·각본(脚本).
2. 인쇄
필기체의 활자.

스크립트 단어를 다음 언어로 번역:

언어 선택



Java™



머니투데이



Scriptlet

→ 자바를 사용해 동적 웹을 구현하기
위한 문법



$\langle \% \sim \% \rangle$

= Scriptlet



```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>

    <title>Title</title>
</head>
<body>
    <%
        int a = 1;
        int b = 2;
        int sum = a + b;
    %>

    <h1>int a 에 저장 된 값은 <%= a %></h1>
    <h1>int b 에 저장 된 값은 <%= int c = 4 %></h1>
    <h1>둘의 합은 <%= sum %></h1>
</body>
</html>
```

HTML 내부에 <% %> 를 쓰고
JAVA 코드를 쓰면 됩니다!!



표현식

<% = ~ % >



```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>

    <title>Title</title>
</head>
<body>
    <%
        int a = 1;
        int b = 2;
        int sum = a + b;
    %>

    <h1>int a 에 저장 된 값은 <%= a %></h1>
    <h1>int b 에 저장 된 값은 <%= int c = 4 %></h1>
    <h1>둘의 합은 <%= sum %></h1>
</body>
</html>
```

특정 값을 바로 '표현' 하고 싶을 땐
<%= %> 를 쓰고 바로 변수를 넣어 줍니다



```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
    <title>Title</title>
</head>
<body>
    <%
        int a = 1;
        int b = 2;
        int sum = a + b;
    %>

    <h1>int a 에 저장 된 값은 <%= a %></h1>
    <h1>int b 에 저장 된 값은 <%= int c = 4 %></h1>
    <h1>둘의 합은 <%= sum %></h1>
</body>
</html>
```

단, 표현식 내부에는 변수만 쓸 수 있으며
요런 자바 문법은 사용이 불가능 합니다~!



아까 코드에서 언급하지 않은
스크립틀릿이 존재 합니다! 어디에 있을까요!?



그런데 말입니다



```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
<html>
    <title>Title</title>
</head>
<body>
    <%
        int a = 1;
        int b = 2;
        int sum = a + b;
    %>

    <h1>int a 에 저장 된 값은 <%= a %></h1>
    <h1>int b 에 저장 된 값은 <%= int c = 4 %></h1>
    <h1>둘의 합은 <%= sum %></h1>
</body>
</html>
```

요 친구도 스크립틀릿 입니다!!

지시자(=Directive)



<%@ ~ %>






페이지 지시자

<%@ page ~ %>



요렇게 생겨서 page 란 이름을 달고
jsp 페이지의 설정을 담당합니다!



```
<%@ page contentType="text/html; charset=UTF-8" language="java" %>
```

페이지 지시자가 설정할 수 있는 것의 종류



- language: 사용 언어를 지정 (기본값은 Java)
- import: 자바 클래스 또는 패키지를 임포트
- contentType: 콘텐츠 유형을 설정 (예: text/html, text/xml)
- session: 세션을 사용할지 여부를 설정 (true 또는 false)
- errorPage: 오류 발생 시 이동할 페이지를 설정
- isErrorPage: 현재 페이지가 오류 페이지인지 여부를 설정



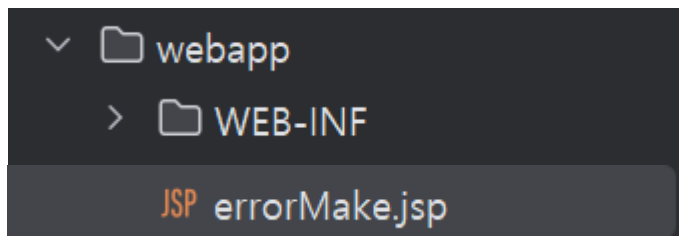
서블릿에서 구현한 Dto 클래스를 импорт

```
<%@ page import="org.example.kbjspServletlecture.PostDto" %>
<%@ page contentType="text/html; charset=UTF-8" language="java" pageEncoding="UTF-8"%>
<%
    PostDto post = (PostDto) request.getAttribute("post");
%>
```

페이지 콘텐츠 유형 및 인코딩,
스크립틀렛의 언어 설정 등등



페이지 지시자로 간편한 에러 처리



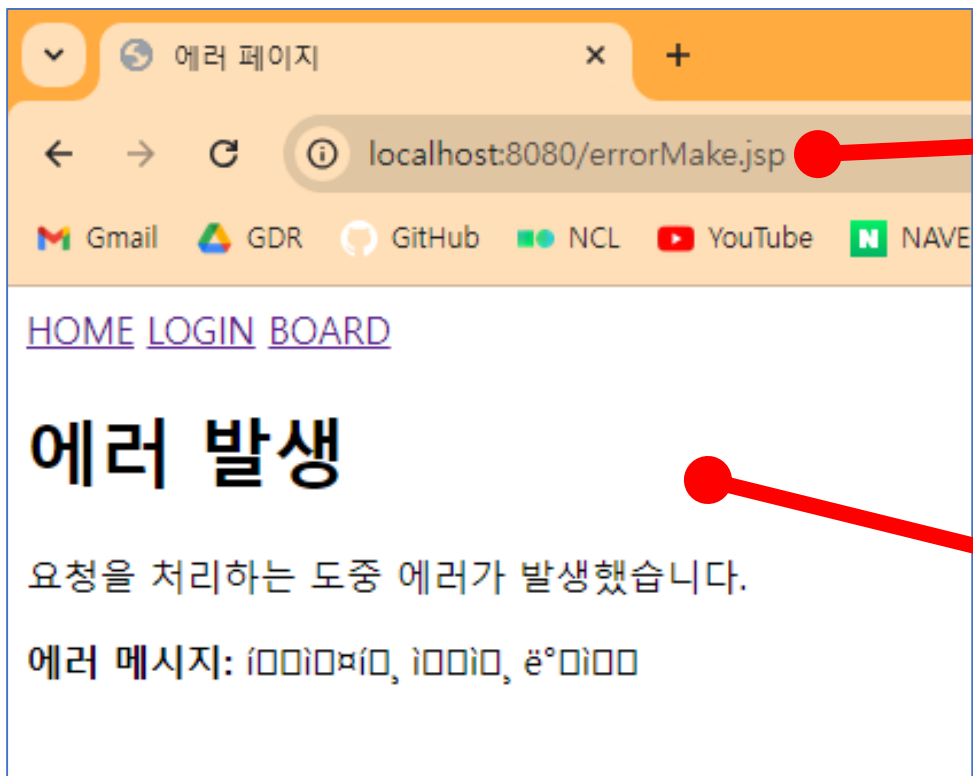
jsp 페이지에서 예외가 발생하면
해당 설정 값에 지정한 페이지로 자동으로
이동(=redirect) 처리가 됩니다!

```
<%@ page errorPage="error.jsp" %>
<%
// 에러를 강제로 발생시키기 위해 사용
if (true) {
    throw new RuntimeException("테스트 예외 발생");
}
%>
```

▼ webapp
 > WEB-INF
 JSP error.jsp

해당 페이지는 에러 처리 페이지임을 표기

```
<%@ page contentType="text/html; charset=UTF-8" language="java" isErrorPage="true" %>
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>에러 페이지</title>
</head>
<body>
  <%@ include file="header.jsp" %>
  <h1>에러 발생</h1>
  <p>요청을 처리하는 도중 에러가 발생했습니다.</p>
  <p><b>에러 메시지:</b> <%= exception.getMessage() %></p>
</body>
</html>
```



errorMake.jsp 페이지에 접속하면
예외가 바로 발생하므로
error.jsp 파일로 리다이렉트!

error.jsp 페이지가 바로 뜨는 것을
확인 가능



include 지시자

taglib 지시자

다른 지시자에 대해서도 한번 알아볼까요?





해당 내용은 앞으로
게시판을 구현하면서
필요할 때 구현하면서
설명하겠습니다!



그럼 일단 CRUD 의 기본인 게시판을 구현 해 봅시다!





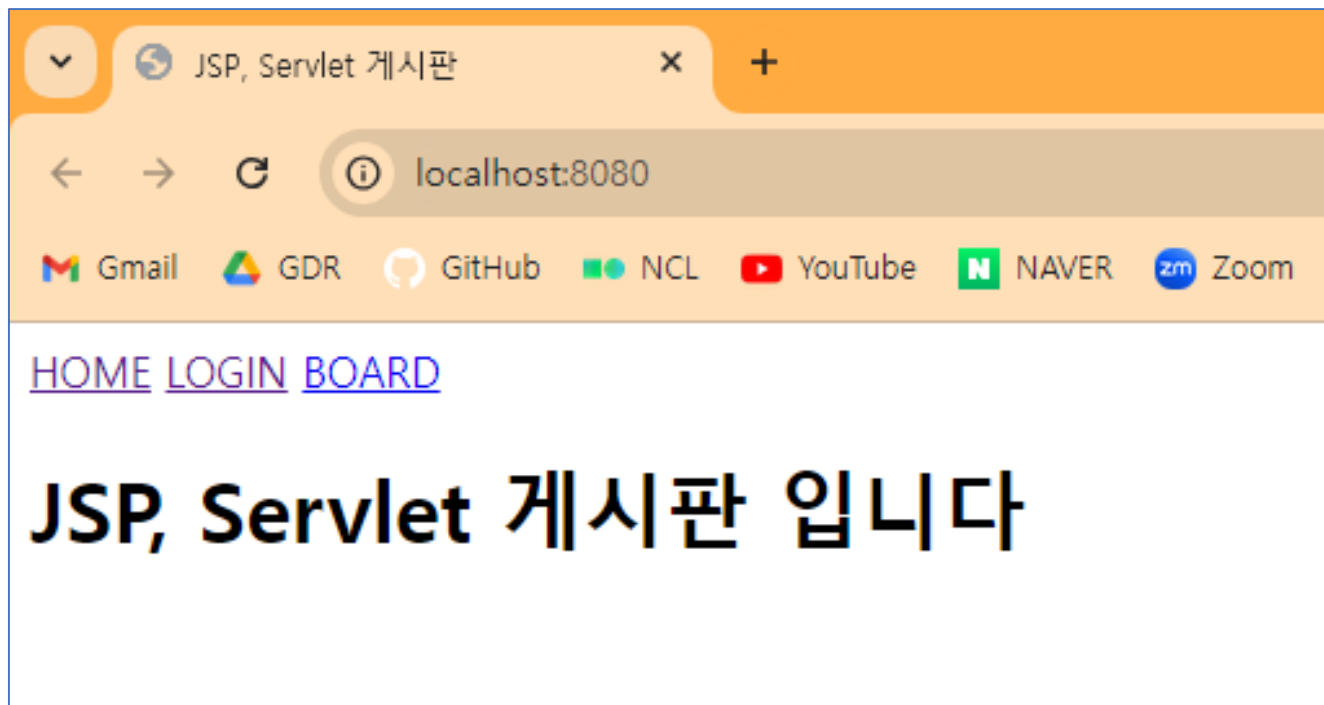
시작 페이지!

index.jsp

```
<%@ page contentType="text/html; charset=UTF-8" language="java" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
    <title>JSP, Servlet 게시판</title>
</head>
<body>
    <header>
        <a href="/">HOME</a>
        <a href="login.jsp">LOGIN</a>
        <a href="board.jsp">BOARD</a>
    </header>
    <h1>JSP, Servlet 게시판 입니다</h1>
</body>
</html>
```

간단한 기능을 위해서
index.jsp 페이지를 꾸며 봅시다!







로그인 페이지!

login.jsp

```
<body>
  <header>
    <a href="/">HOME</a>
    <a href="login.jsp">LOGIN</a>
    <a href="board.jsp">BOARD</a>
  </header>
  <h1>로그인</h1>
  <form method="get" action="login">
    <label for="username">아이디 :</label>
    <input type="text" id="username" name="username" required>
    <br>
    <label for="password">비밀번호 :</label>
    <input type="password" id="password" name="password" required>
    <br>
    <button type="submit">로그인</button>
  </form>
</body>
```

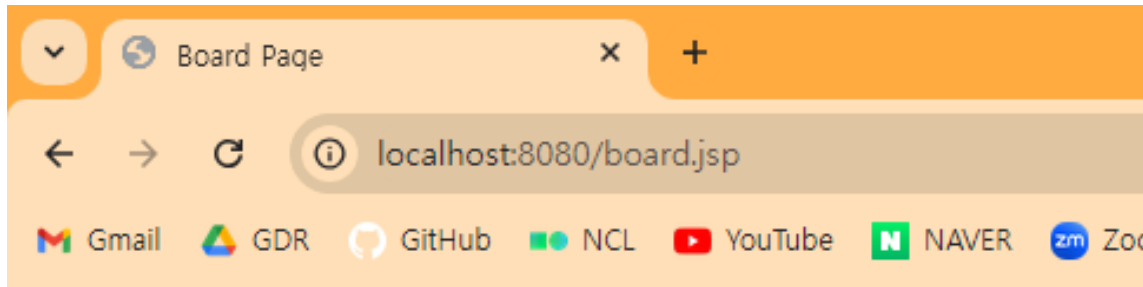
form 을 통해 서블릿에
username 과 password 를 전달하여
로그인을 요청하는 페이지!



초단간 실습, 게시판 페이지 만들기



- 게시판 목록을 보여줄 board.jsp 페이지를 만들어 주세요
- board.jsp 파일은 <h1> 태그로 '게시판 페이지 입니다' 를 출력 합니다
- 앞서 작성한 <haeder /> 가 들어가도록 만들어 주세요!

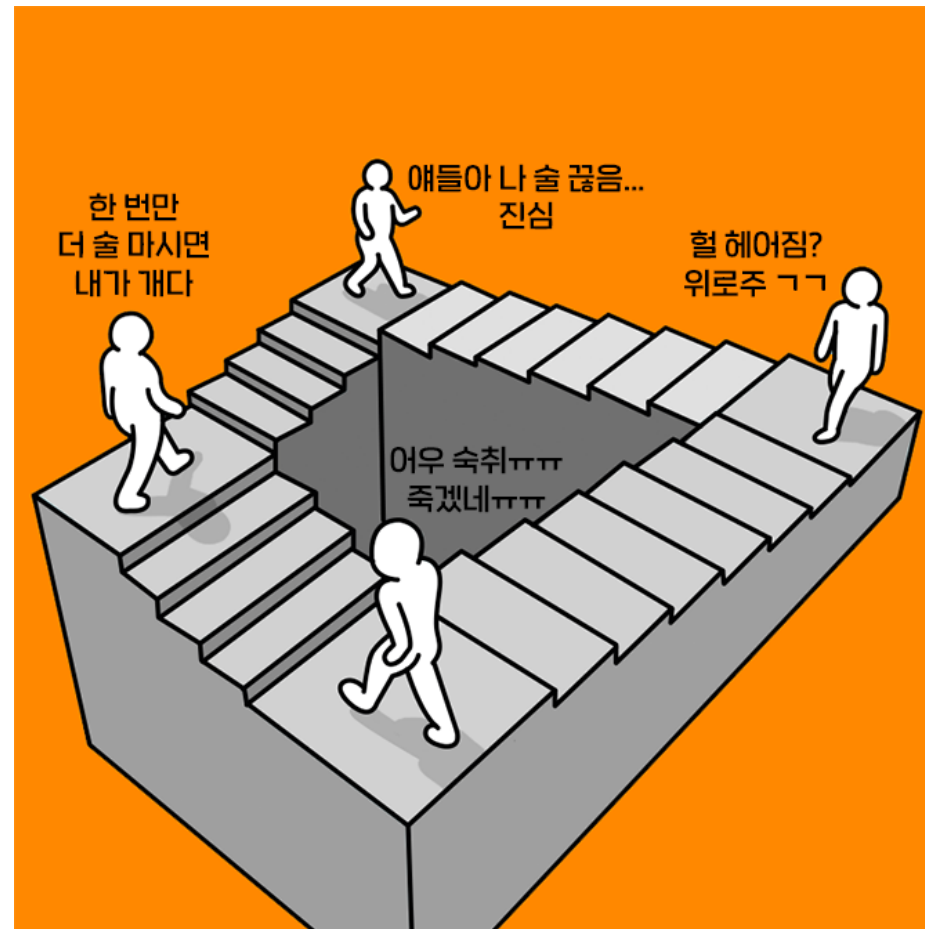


[HOME](#) [LOGIN](#) [BOARD](#)

게시판 페이지 입니다



반복이 발생!





그런데 말입니다

모든 페이지에는 `<haeder />` 가 들어 갈텐데
이거 매번 넣어줘야 할까요!?

그런데 `<header />` 가 갑자기 변경 된다면!?

그럼 뭘 써야 할까요!?





include 지시자

<%@ include ~ %>

<body>

<h1>로그인</h1>

<form method="post" action="login">

<label for="username">아이디 :</label>

<input type="text" id="username" name="username" required>

우리가 원하는 것은 딱 요부분에
특정 HTML 코드를 넣고 싶습니다!

→ 요 때 include 지시자를 사용!

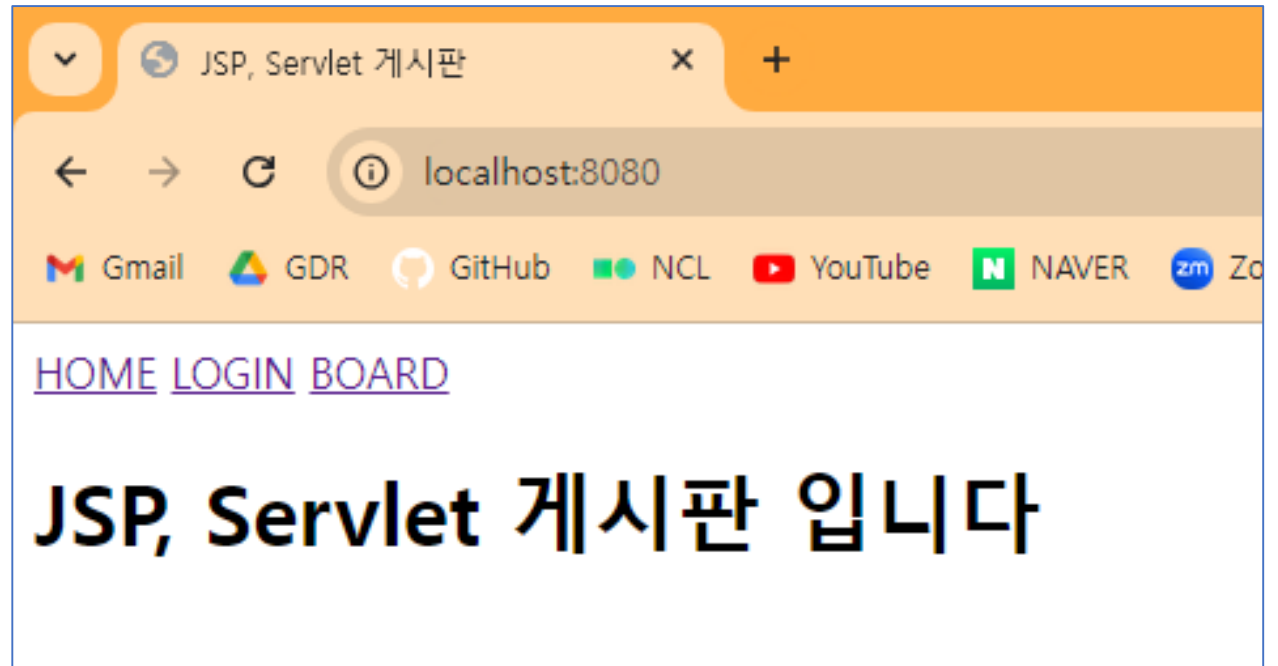


```
<%@ page contentType="text/html; charset=UTF-8" language="java" pageEncoding="UTF-8"%>
<header>
  <a href="/">HOME</a>
  <a href="login.jsp">LOGIN</a>
  <a href="board.jsp">BOARD</a>
</header>
```

우리가 원하는 HTML 코드를 담은
header.jsp 파일을 생성

```
<head>
  <title>JSP, Servlet 게시판</title>
</head>
<body>
  <%@ include file="header.jsp" %>
  <h1>JSP, Servlet 게시판 입니다</h1>
</body>
</html>
```

include 지시자를 이용하여
header.jsp 의 HTML 코드를
index.jsp 페이지에 적용!





```
<%@ page contentType="text/html; charset=UTF-8" language="java" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Login Page</title>
</head>
<body>
    <%@ include file="header.jsp" %>
    <h1>로그인</h1>
    <form method="get" action="login">
        <label for="username">아이디 :</label>
        <input type="text" id="username" name="username" required>
        <br>
        <label for="password">비밀번호 :</label>
        <input type="password" id="password" name="password" required>
        <br>
        <button type="submit">로그인</button>
    </form>
</body>
</html>
```

include 지시자를 이용하여
header.jsp 의 HTML 코드를
login.jsp 페이지에도 적용!

⌵

🌐 Login Page

✕

+

⬅

➡

🔄

📁 localhost:8080/login.jsp

📧 Gmail

📁 GDR

🔄 GitHub

🟢 NCL

📺 YouTube

🟢 NAVER

2

[HOME](#) [LOGIN](#) [BOARD](#)

로그인

아이디 :

비밀번호 :

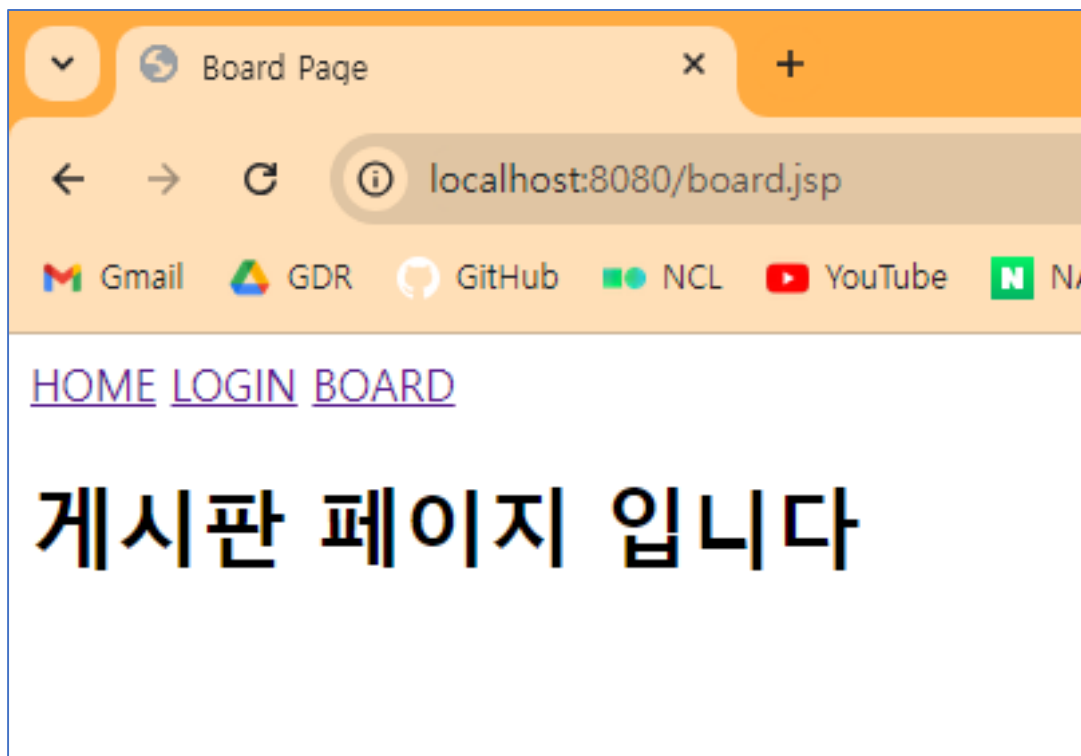
로그인



```
<head>
  <title>Board Page</title>
</head>
<body>
  <%@ include file="header.jsp" %>
  <h1>게시판 페이지 입니다</h1>
</body>
</html>
```

include 지시자를 이용하여
header.jsp 의 HTML 코드를
board.jsp 페이지에도 적용!







주소를 바꿔 봅시다

Login Page

localhost:8080/login.jsp

Gmail GDR GitHub NCL YouTube

[HOME](#) [LOGIN](#) [BOARD](#)

로그인

아이디 :

비밀번호 :

로그인


지금은 주소 값으로 파일명이 나오고 있습니다!
이걸 /login 처럼 주소로 변경해 봅시다!



Servlets



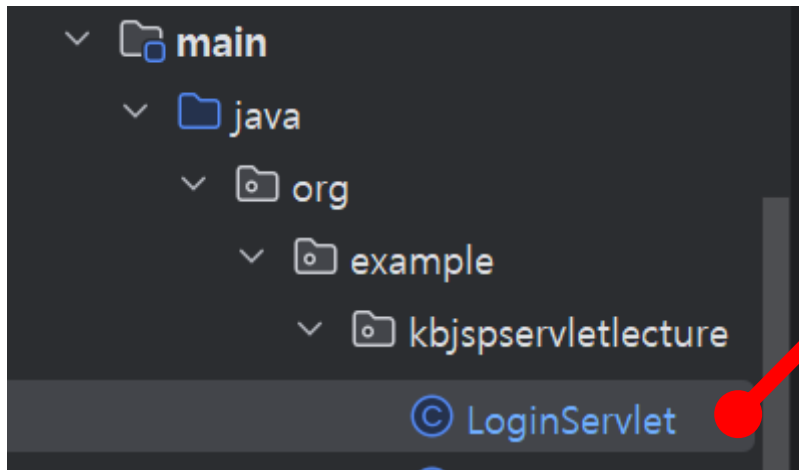
이 몸, 등장

 New Servlet ✕

JAVAAEE TYPE:

URL MAPPING:

ClassName:



로그인을 처리할 LoginServlet 만들기!

/login 주소로 요청이 오면
login.jsp 파일을 서빙 하도록 변경!



```
@WebServlet("/login")  👤 kdtTetz *
public class LoginServlet extends HttpServlet {
    @Override  3 usages  👤 kdtTetz *
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        // /login 주소로 요청이 오면 로그인 페이지 파일을 제공
        RequestDispatcher dispatcher = request.getRequestDispatcher(s: "login.jsp");
        dispatcher.forward(request, response);
    }

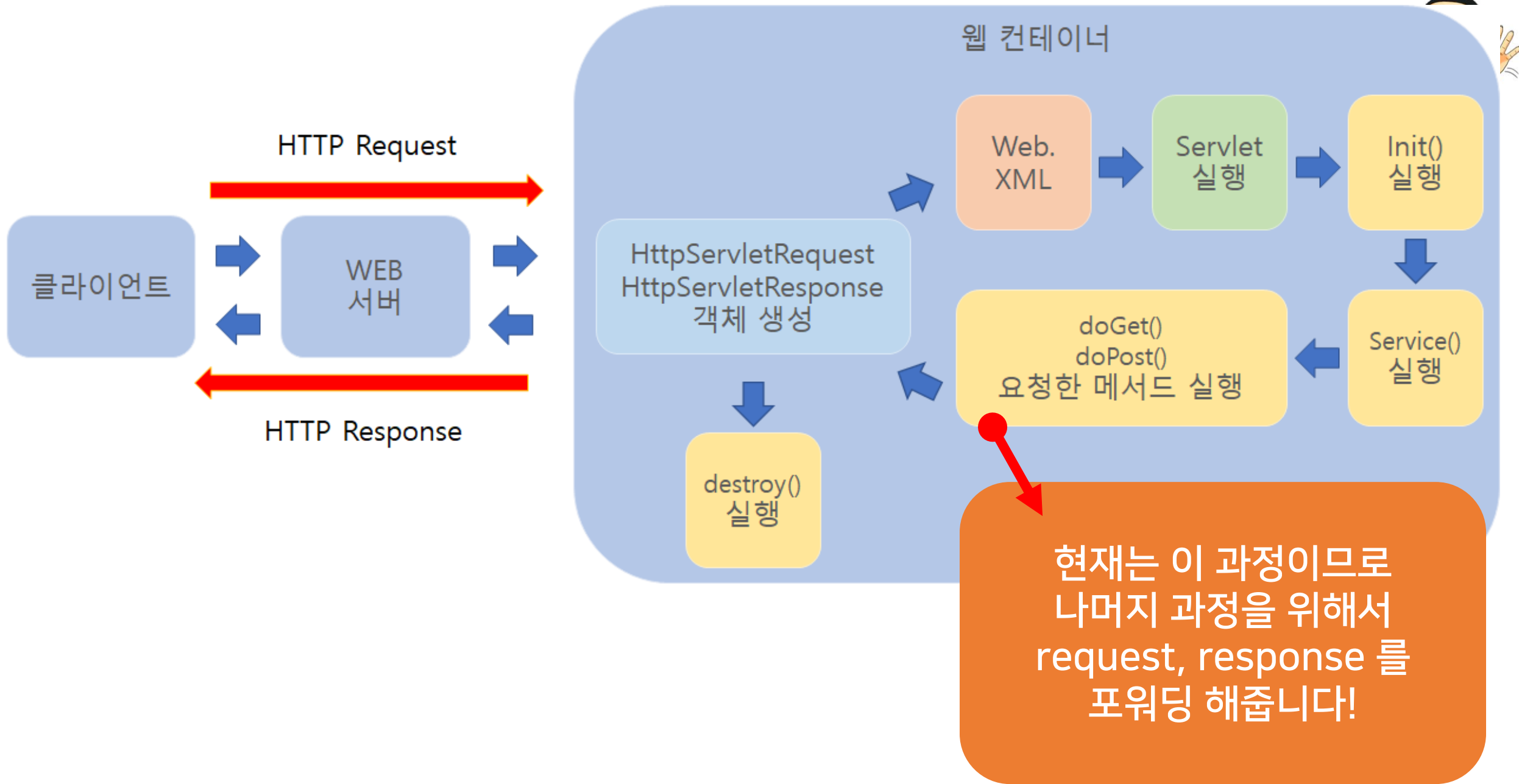
    @Override  no usages  👤 kdtTetz *
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException {
        doGet(request, response);
    }
}
```



```
@WebServlet("/login")  👤 kdtTetz *  
public class LoginServlet extends HttpServlet {  
    @Override  3 usages  👤 kdtTetz *  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws S  
        // /login 주소로 요청이 오면 로그인 페이지 파일을 제공  
        RequestDispatcher dispatcher = request.getRequestDispatcher(s: "login.jsp");  
        dispatcher.forward(request, response);  
    }  
  
    @Override  no usages  👤 kdtTetz *  
    protected void doPost(HttpServletRequest request  
        doGet(request, response);  
    }  
}
```

RequestDispatcher 를 사용하면
특정 파일을 서빙할 수 있습니다!

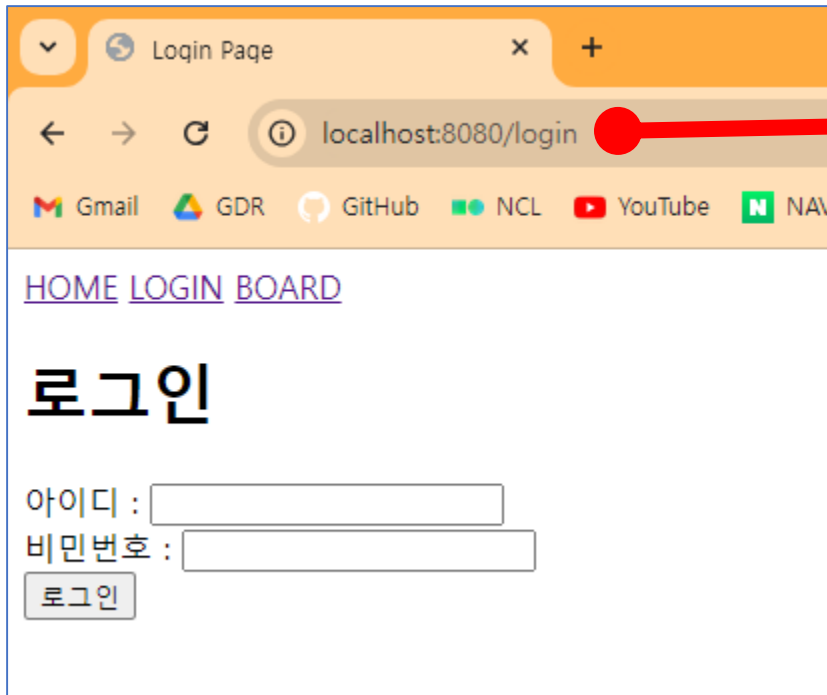
그리고, 해당 request 와 response 는
아직 해야할 과정이 남아 있기 때문에
다시 forwarding 해줍니다!



```
<%@ page contentType="text/html;char
<header>
  <a href="/">HOME</a>
  <a href="/login">LOGIN</a>
  <a href="/board.jsp">BOARD</a>
</header>
```

헤더 파일의 login 요청 주소를 변경





Browser: Login Page
URL: localhost:8080/login

HOME LOGIN BOARD

로그인

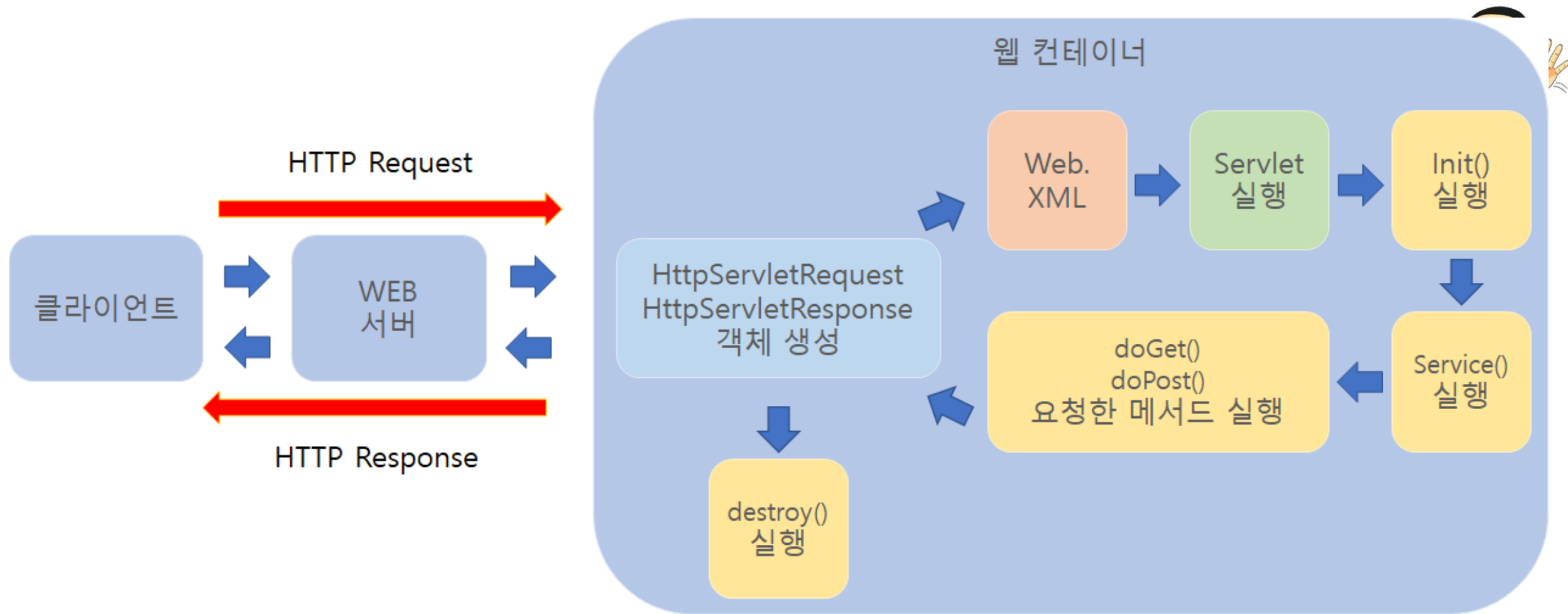
아이디 :

비밀번호 :

/login 으로 주소 요청이 들어가면
LoginSevelt 이 먼저 요청을 처리하고
login.jsp 파일을 서빙!

→ login.jsp 파일을 요청한 것과 동일한 결과!

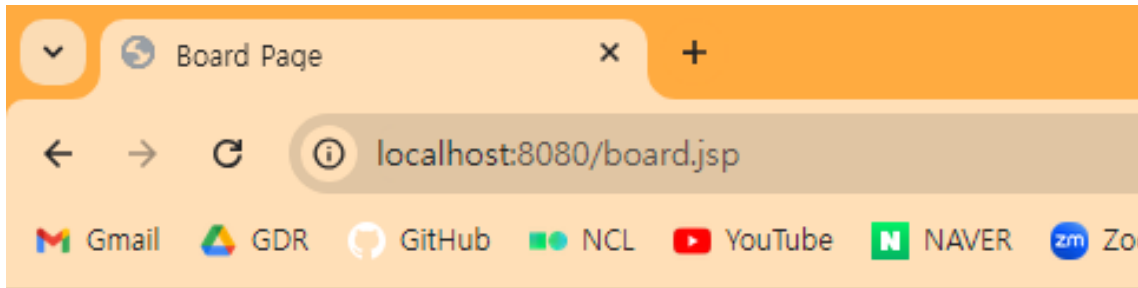




실습, 게시판 페이지의 주소 요청 값 변경하기



- 현재 board.jsp 로 되어있는 게시판 페이지 요청을 /board 로 변경해 봅시다!



[HOME](#) [LOGIN](#) [BOARD](#)

게시판 페이지 입니다

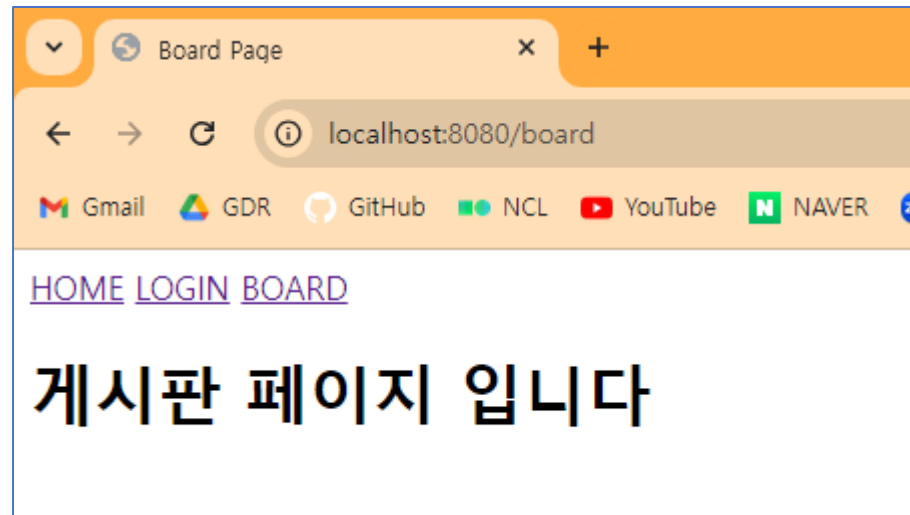
- BoardServlet 클래스를 만들어 /board 요청이 들어오면 board.jsp 파일을 서빙하는 기능을 추가해 봅시다!

실습, 게시판 페이지의 주소 요청 값 변경하기



- 아래와 같이 헤더의 주소 요청을 변경하고, BOARD 링크를 클릭 했을 때 아래와 같이 게시판 페이지가 뜨면 됩니다!

```
<%@ page contentType="text/html; charset=utf-8"%>
<html>
<head>
<title>Board Page</title>
<meta charset="utf-8"%>
<script src="js/jquery-3.6.0.min.js"></script>
<script src="js/board.js"></script>
</head>
<body>
<div class="container">
<div class="row">
<div class="col">
<a href="/">HOME</a>
<a href="/login">LOGIN</a>
<a href="/board">BOARD</a>
</div>
</div>
</body>
</html>
```

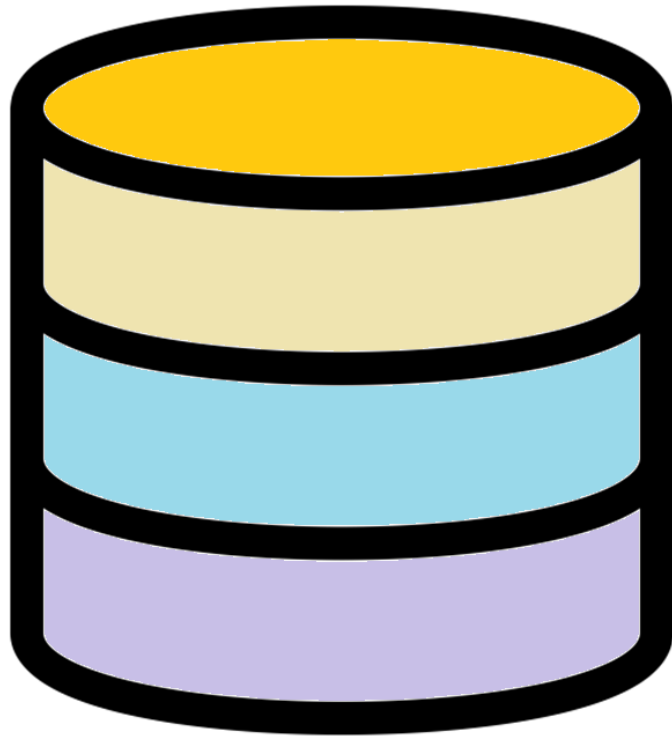




대뜸

로그인 구현!

로그인을 하려면 뭐가 필요하죠!?



**DB
/SQL**

하지만 우리가 틀렸다면

지금 옳다고 믿는 것에 대한
베딱한 질문

책 클로스터먼 지음 윤태경 옮김

But We're Wrong!





```
CREATE DATABASE board;
```

```
USE board;
```

```
CREATE TABLE users (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,  
    username VARCHAR(50) NOT NULL UNIQUE,  
    password VARCHAR(50) NOT NULL
```

```
);
```

```
INSERT INTO users (username, password) VALUES ('tetz', '1234');
```

```
SELECT * FROM users;
```


Result Grid



Filter Rows:

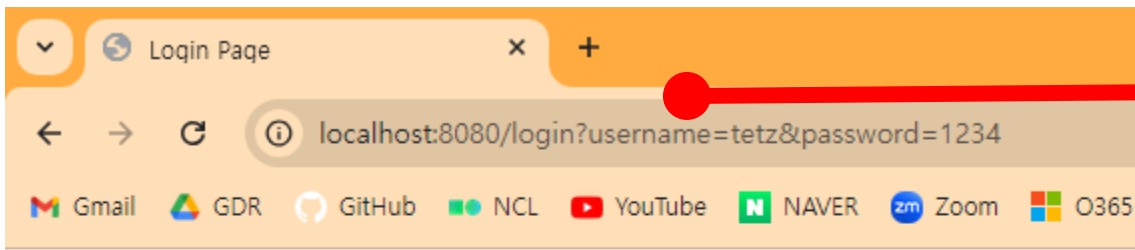
	id	username	password
▶	1	tetz	1234
●	NULL	NULL	NULL



요 페이지에서 로그인을 시도하면!?



```
<body>
  <%@ include file="header.jsp" %>
  <h1>로그인</h1>
  <form method="get" action="login">
    <label for="username">아이디 :</label>
    <input type="text" id="username" name="username" required>
    <br>
    <label for="password">비밀번호 :</label>
    <input type="password" id="password" name="password" required>
    <br>
    <button type="submit">로그인</button>
  </form>
</body>
```



그런데 ? 뒤에 있는 애는 뭐죠? 아시는 분?

[HOME](#) [LOGIN](#) [BOARD](#)

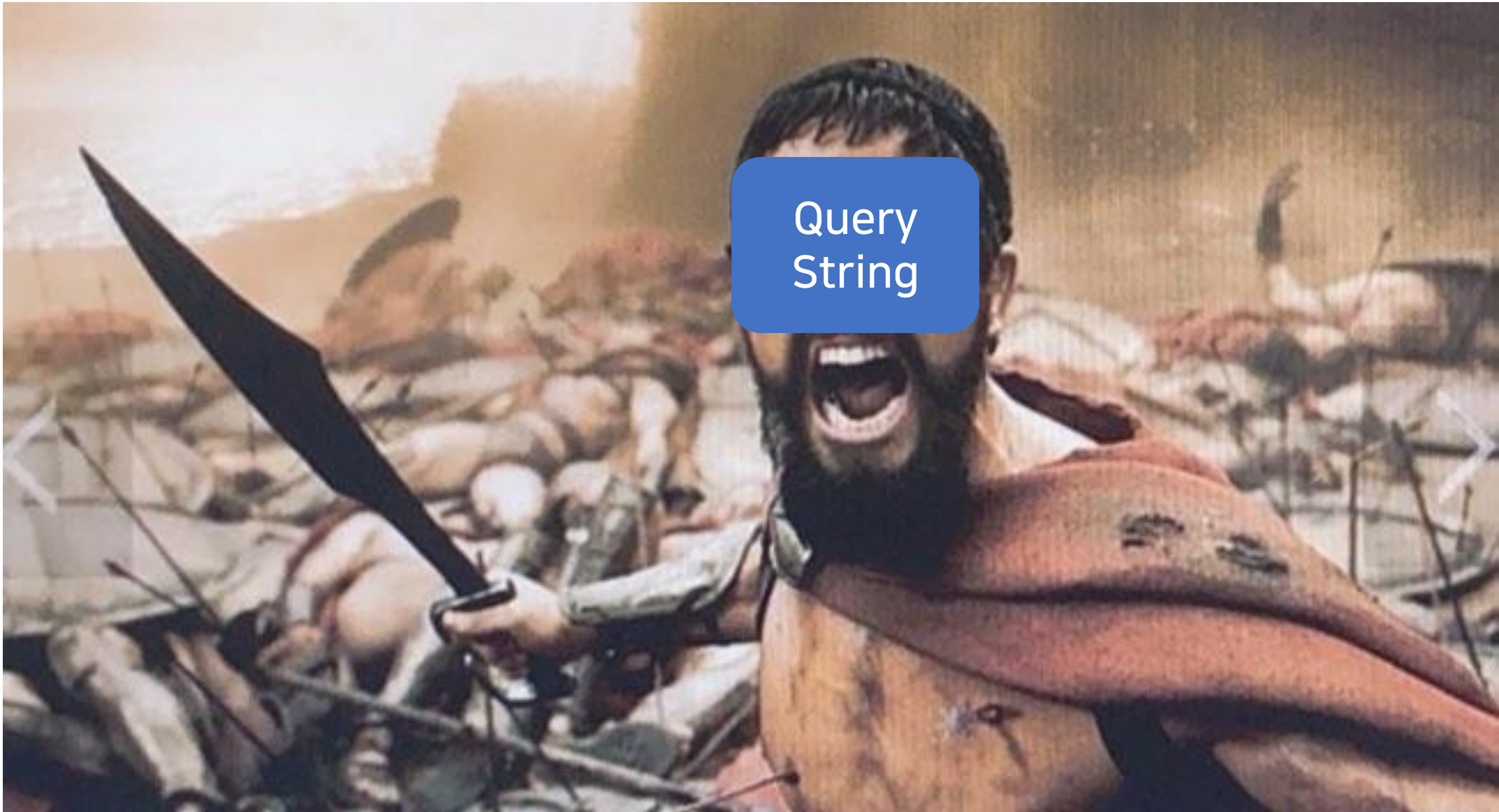
로그인

아이디 :

비밀번호 :

action 속성의 값이 login 이므로
/login 으로 요청을 보내게 됩니다!

현재 /login 주소 값에 대한 처리는
login.jsp 파일을 서빙하는 것이므로
똑같이 로그인 페이지가 뜹니다!



Query
String



QueryString!



localhost:8080/login?username=tetz&password=1234



자세히 보면 아까 여러분들이 보냈던
데이터랑 똑같습니다!!



localhost:8080/login?username=tetz&password=1234



```
<body>
```

```
<%@ include file="header.jsp" %>
```

```
<h1>로그인</h1>
```

```
<form method="get" action="login">
```

```
  <label for="username">아이디 :</label>
```

```
  <input type="text" id="username" name="username" required>
```

```
  <br>
```

```
  <label for="password">비밀번호 :</label>
```

```
  <input type="password" id="password" name="password" required>
```

```
  <br>
```

```
  <button type="submit">로그인</button>
```

```
</form>
```

```
</body>
```

요걸 보낼 때 서버에게
정보 전달을 위해서 사용하는 것 입니다!



클라이언트 (Client)

Request



Response

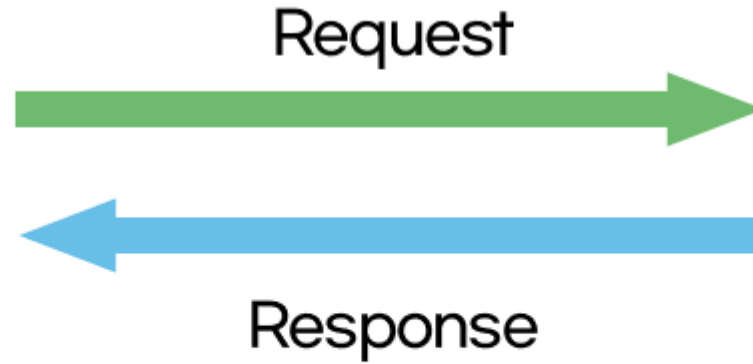


서버 (Server)

그럼 서버 입장(=Servlet) 에서
확인해 봅시다!



클라이언트 (Client)



서버 (Server)



```
@WebServlet("/login")  👤 kdtTetz *
public class LoginServlet extends HttpServlet {
    @Override  👤 kdtTetz *
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        String username = request.getParameter(s: "username");
        String password = request.getParameter(s: "password");

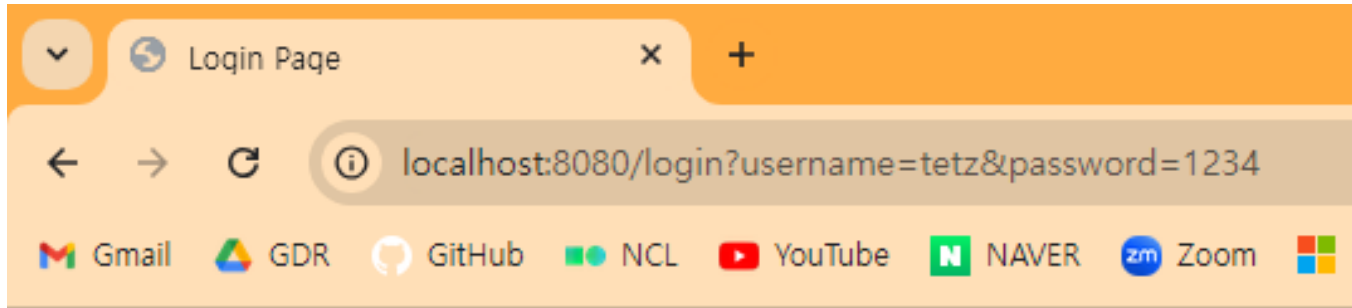
        System.out.println("##### 로그인 요청!! #####");
        System.out.println("username = " + username);
        System.out.println("password = " + password);

        // /login 주소로 요청이 오면 로그인 페이지 파일을 제공
        RequestDispatcher dispatcher = request.getRequestDispatcher(s: "login.jsp");
        dispatcher.forward(request, response);
    }
}
```

request 에서
getParameter 를 사용해서
각각의 정보를 받아오기!



```
##### 로그인 요청!! #####  
username = null  
password = null
```



[HOME](#) [LOGIN](#) [BOARD](#)

```
##### 로그인 요청!! #####  
username = tetz  
password = 1234
```



Rank Transform of an Array

example

$[40, 10, 20, 30] \rightarrow [4, 1, 2, 3]$ ^{OK}

"Edge Case"

배열 empty? OK

중복 숫자 허용? OK

$\hookrightarrow [100, 100, 100] \rightarrow [1, 1, 1]$ ^{OK}

배열의 최대 길이는? $0 \leq \text{length} \leq 10^5$

배열에 포함된 숫자의 범위는? $-10^9 \leq \text{num} \leq 10^9$
 \hookrightarrow signed 32bit

$31, 12, 28, 9, 100, 56, 80, 51, 12$

$5, 3, 4, 2, 8, 6, 1, 1, 3$ ^{OK}

```
vector<int> solve (vector<int>& arr) {  
    vector<int> tmp (arr);  
    map<int, int> rank;   
    for (int & item : tmp) {  
        rank.emplace (item, rank.size()+1);  
    }  
    for (int & item : arr) {  
        item = rank[item];  
    }  
    return arr;  
}
```

$O(n) + O(n) = O(n)$ ^{time complexity}

$O(n) + O(n) = O(n)$ ^{space complexity}



ID 랑 PASSWORD 가
저렇게 주소창에 뜨는게 맞을까요!?

→ 그럼 뭘 쓰면 되나요!?



그란데 말입니다



```
<body>
```

```
<%@ include file="header.jsp" %>
```

```
<h1>로그인</h1>
```

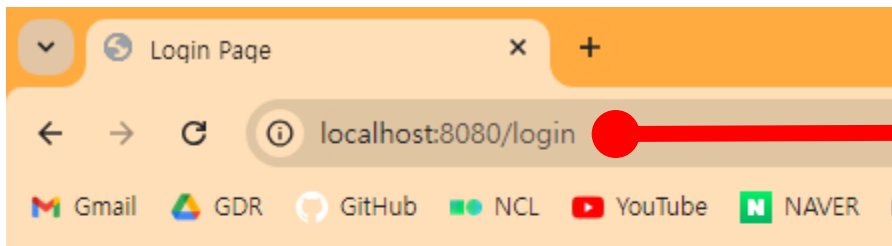
```
<form method="post" action="login">
```

```
  <label for="username">아이디 :</label>
```

```
  <input type="text" id="username" name="username" required>
```

메소드를 POST 로 변경!





로그인을 눌러도
주소창에 더 이상 쿼리가 뜨지 않습니다!



[HOME](#) [LOGIN](#) [BOARD](#)

로그인

아이디 :

비밀번호 :



그란데 말입니다

```
##### 로그인 요청!! #####  
username = tetz  
password = 1234
```

서버 로그에는 요게 찍히네요!?

왜 그런지 아시는 분!?



JDBC

두둥등장!



build.gradle 파일

MySQL 연결용 드라이버 라이브러리
추가

```
dependencies {  
    compileOnly('javax.servlet:javax.servlet-api:4.0.1')  
    implementation 'mysql:mysql-connector-java:8.0.30'  
  
    testImplementation("org.junit.jupiter:junit-jupiter-api:${junitVersion}")  
    testRuntimeOnly("org.junit.jupiter:junit-jupiter-engine:${junitVersion}")  
}
```

```
implementation 'mysql:mysql-connector-java:8.0.30'
```



```
@WebServlet("/login")  👤 kdtTetz *  
public class LoginServlet extends HttpServlet {  
    private static final String JDBC_URL = "jdbc:mysql://localhost:3306/board";  
    private static final String JDBC_USER = "root";  no usages  
    private static final String JDBC_PASSWORD = "1234";  no usages  
}
```



JDBC 연결을 위한 정보 문자열 선언!

request 로 부터 username 과 password 정보를 가져오기!



```
@Override no usages kdtTetz *
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
    String username = request.getParameter(s: "username");
    String password = request.getParameter(s: "password");

    boolean isLoginSuccess = false;

    try {
        Class.forName(className: "com.mysql.cj.jdbc.Driver");
        String sql = "SELECT * FROM users WHERE username = ? AND password = ?";
```

로그인 성공 여부를 저장할 변수

MySQL 연결용 드라이버 로드!

로그인을 위한 SQL 구문
작성

```
@Override no usages kdtTetz *
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
    String username = request.getParameter(s: "username");
    String password = request.getParameter(s: "password");

    boolean isLoginSuccess = false;

    try {
        Class.forName(className: "com.mysql.cj.jdbc.Driver");
        String sql = "SELECT * FROM users WHERE username = ? AND password = ?";
```



```
try {
    Class.forName(className: "com.mysql.cj.jdbc.Driver");
    String sql = "SELECT * FROM users WHERE username = ? AND password = ?";

    try (Connection conn = DriverManager.getConnection(JDBC_URL, JDBC_USER, JDBC_PA
        PreparedStatement pstmt = conn.prepareStatement(sql)) {

        pstmt.setString(parameterIndex: 1, username);
        pstmt.setString(parameterIndex: 2, password);

        try (ResultSet rs = pstmt.executeQuery()) {
            if (rs.next()) {
                isLoginSuccess = true;
            }
        }
    }
}
```

필요 정보를 Driver 에 전달하여
MySQL 에 연결



```
try {  
    Class.forName(className: "com.mysql.cj.jdbc.Driver");  
    String sql = "SELECT * FROM users WHERE username = ? AND password = ?";  
  
    try (Connection conn = DriverManager.getConnection(JDBC_URL, JDBC_USER, JDBC_PA  
        PreparedStatement pstmt = conn.prepareStatement(sql)) {  
  
        pstmt.setString(parameterIndex: 1, username);  
        pstmt.setString(parameterIndex: 2, password);  
  
        try (ResultSet rs = pstmt.executeQuery()) {  
            if (rs.next()) {  
                isLoginSuccess = true;  
            }  
        }  
    }  
}
```

PreparedStatement 준비

request 로 부터 받은
username 과 password 를
사용해서

PreparedStatement 완성!



```
try {
    Class.forName(className: "com.mysql.cj.jdbc.Driver");
    String sql = "SELECT * FROM users WHERE username = ? AND password = ?";

    try (Connection conn = DriverManager.getConnection(JDBC_URL, JDBC_USER, JDBC_PA
        PreparedStatement pstmt = conn.prepareStatement(sql)) {

        pstmt.setString(parameterIndex: 1, username);
        pstmt.setString(parameterIndex: 2, password);


        try (ResultSet rs = pstmt.executeQuery()) {
            if (rs.next()) {
                isLoginSuccess = true;
            }
        }
    }
}
```

PreparedStatement 수행!
데이터를 하나라도 찾았다면!?
→ 로그인 성공!!



```
try (ResultSet rs = pstmt.executeQuery()) {  
    if (rs.next()) {  
        isLoginSuccess = true;  
    }  
}  
  
} catch (Exception e) {  
    e.printStackTrace();  
}
```

다양한 예외를 처리하기 위해
Exception 예외 객체로 catch!



로그인 성공 여부를 저장하는
isLoggedIn 변수를 통해
로그인 성공 여부를 체크하기!

```
if (isLoggedIn) {  
    request.setAttribute(s: "username", username);  
    RequestDispatcher dispatcher = request.getRequestDispatcher(s: "welcome.jsp");  
    dispatcher.forward(request, response);  
} else {  
    response.sendRedirect(s: "loginFailed.jsp");  
}
```

jsp 페이지에 데이터 전달이
가능한 request 객체에
setAttribute 메서드를 이용하여
username 이라는 키에
사용자의 id 값을 저장하여 전달!

```
if (isLoginSuccess) {  
    request.setAttribute(s: "username", username);  
    RequestDispatcher dispatcher = request.getRequestDispatcher(s: "welcome.jsp");  
    dispatcher.forward(request, response);  
} else {  
    response.sendRedirect(s: "loginFailed.jsp");  
}
```



```
if (isLoginSuccess) {  
    request.setAttribute(s: "username", username);  
    RequestDispatcher dispatcher = request.getRequestDispatcher(s: "welcome.jsp");  
    dispatcher.forward(request, response);  
} else {  
    response.sendRedirect(s: "loginFailed.jsp");  
}
```

RequestDispatcher 를 사용해서
welcome 페이지로 이동 시키기!

request 와 reponse 는 아직
이동이 덜 끝났으므로
고대로 전달하기!

welcome.jsp 파일



```
<%@ page contentType="text/html; charset=UTF-8" language="java" pageEncoding="UTF-8"%>
<html>
<head>
  <title>Welcome</title>
</head>
<%@ include file="header.jsp" %>
<body>
  <h1>로그인 성공!</h1>
  <h2>환영합니다! <%= request.getAttribute("username") %> 님</h2>
  <h2>환영합니다! ${username} 님</h2>
</body>
<html>
```

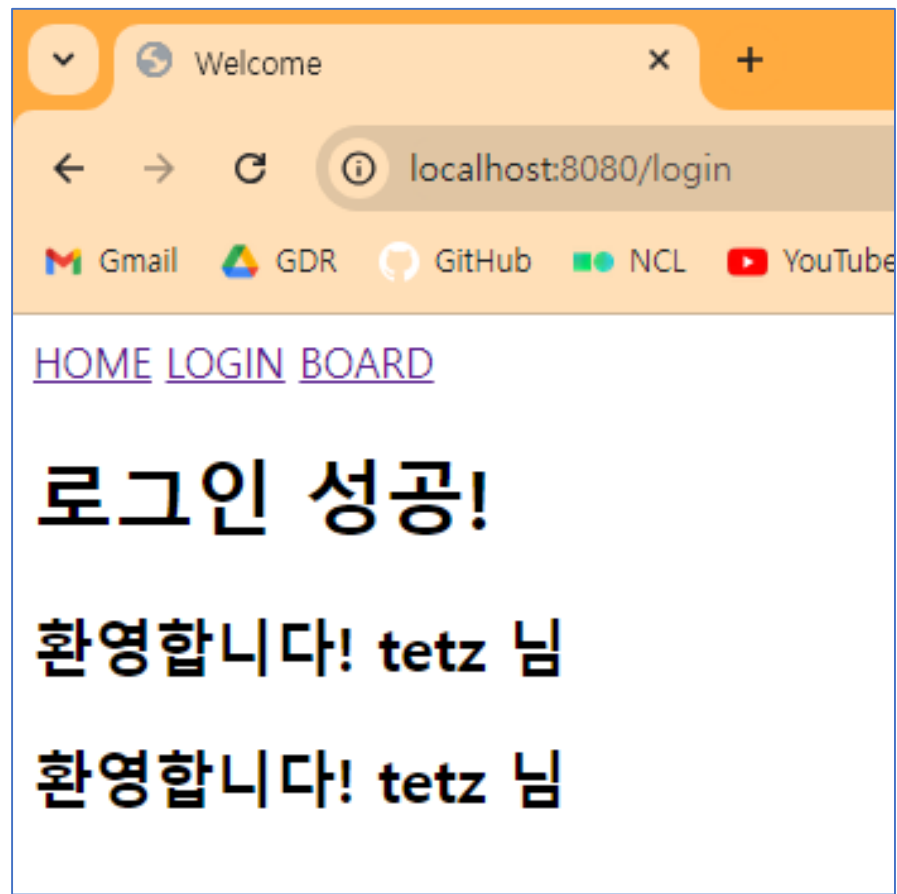
request 객체에 담겨온
username 의 값을 꺼내서 출력!

EL(Expression Language) 를
이용해서 표현 가능!

EL은 다음에 더 깊게 다룰 예정입니다!



자~ 드가자~





```
if (isLoginSuccess) {  
    request.setAttribute(s: "username", username);  
    RequestDispatcher dispatcher = request.getRequestDispatcher(s: "welcome.jsp");  
    dispatcher.forward(request, response);  
} else {  
    response.sendRedirect(s: "loginFailed.jsp");  
}
```

로그인이 실패하면
loginFailed.jsp 파일로 보내기!

loginFailed.jsp 파일



```
<%@ page contentType="text/html; charset=UTF-8" language="java" pageEncoding="UTF-8"%>
<html>
<head>
    <title>Title</title>
</head>
<%@ include file="header.jsp" %>
<body>
    <h1>로그인 실패!</h1>
    <h2>ID 또는 PW 가 틀립니다</h2>
    <a href="login"><button type="button">로그인 페이지로</button></a>
</body>
<html>
```



Login Page

localhost:8080/login

Gmail GDR GitHub NCL YouTube NAVER Zoom

[HOME](#) [LOGIN](#) [BOARD](#)

로그인

아이디 :

비밀번호 :

Welcome

localhost:8080/login

Gmail GDR GitHub NCL YouTube

[HOME](#) [LOGIN](#) [BOARD](#)

로그인 성공!

환영합니다! tetz 님

환영합니다! tetz 님

Title

localhost:8080/loginFailed.jsp

Gmail GDR GitHub NCL YouTube NAVER Zoom

[HOME](#) [LOGIN](#) [BOARD](#)

로그인 실패!

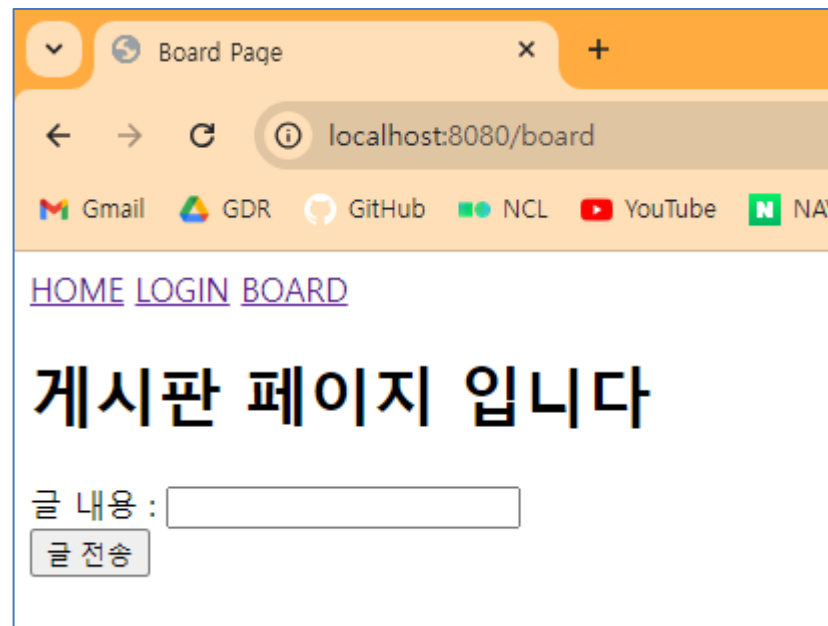
ID 또는 PW 가 틀립니다

실습, 게시판에 글 전달 기능 구현하기!



- board.jsp 파일을 아래와 같이 만들어 주세요

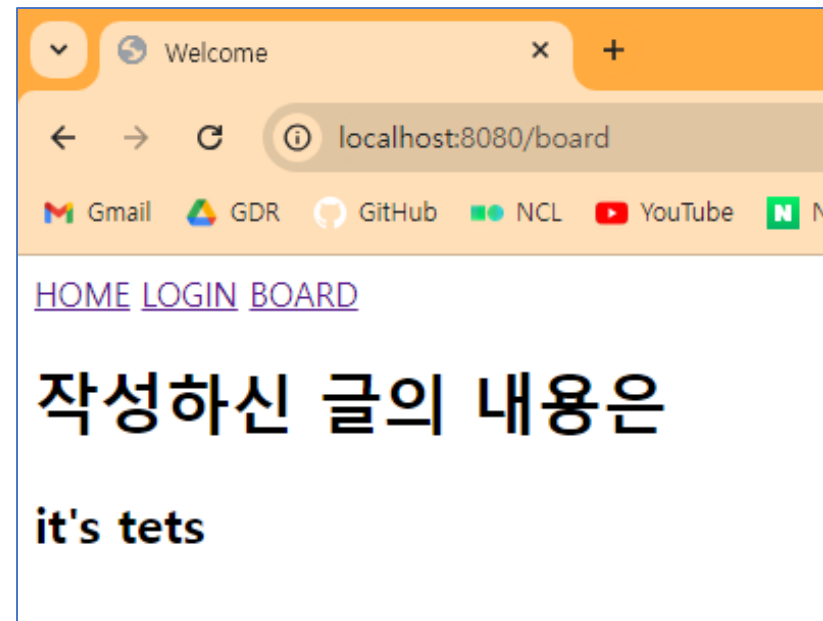
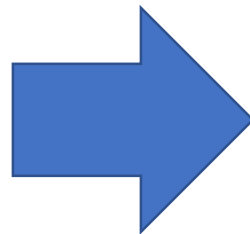
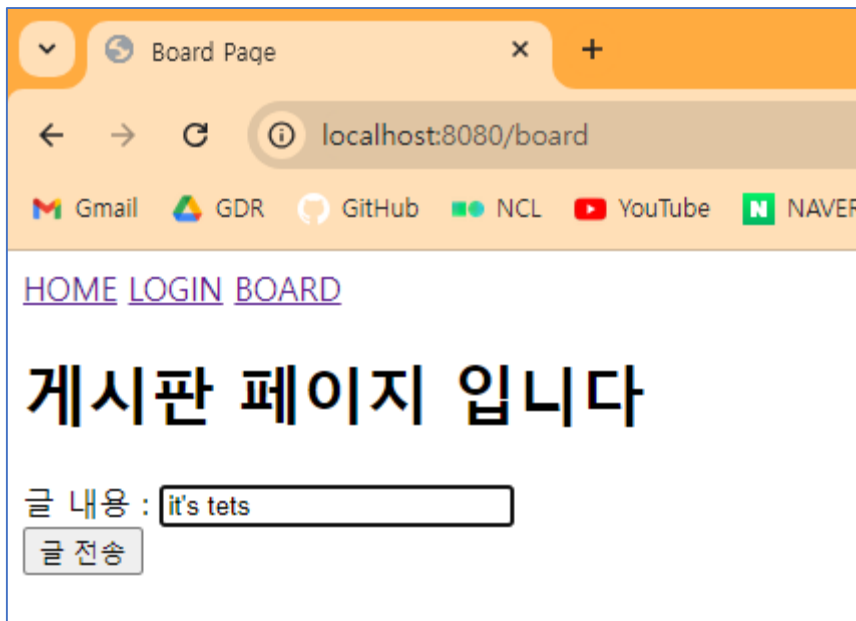
```
<body>
  <%@ include file="header.jsp" %>
  <h1>게시판 페이지 입니다</h1>
  <form method="post" action="board">
    <label for="post">글 내용 :</label>
    <input type="text" id="post" name="post" required>
    <br>
    <button type="submit">글 전송</button>
  </form>
</body>
```



실습, 게시판에 데이터를 보내고 출력하기



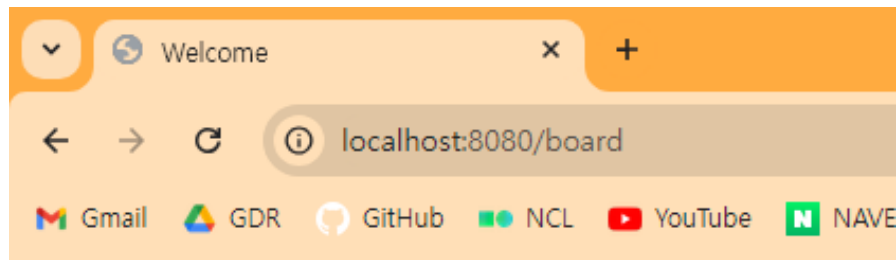
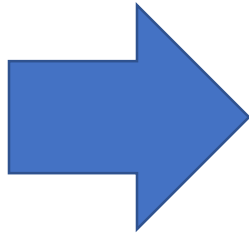
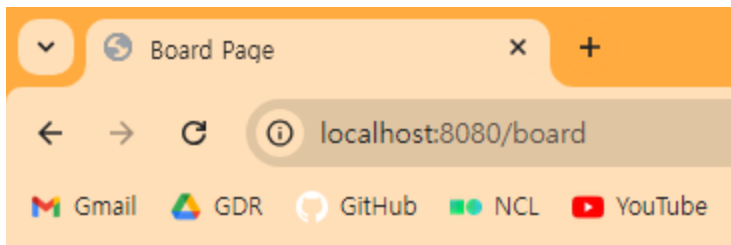
- 이전 실습에서 구현한 BoardServlet 에 POST 방식으로 전달 된, 글 내용을 받아서 boardDetail.jsp 파일에 전달하는 기능을 구현해 주세요
- boardDrtail.jsp 파일은 아래와 같이 출력되면 됩니다!



추가 실습, 한글 처리하기!



- 한글을 지금 한글을 보내면 아래와 같이 깨집니다!



- 오전 강의에서 배운 내용을 바탕으로 한글이 안깨지도록 해주세요!