

class 230127

AJAX

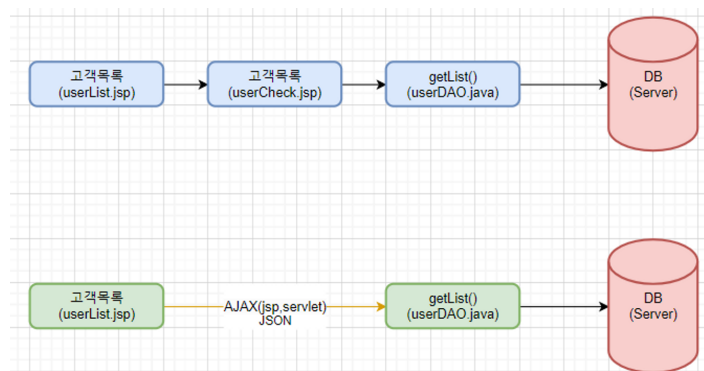
Ajax(Asynchronous JavaScript and XML)는 동적 웹 페이지를 만들기 위한 기법 중 하나로, 웹 페이지 전체를 다시 로딩하지 않고 웹 페이지 일부분만을 갱신할 수 있게 해줌.

등장 배경

- JSP를 통한 사용자 요청에 대한 결과 페이지를 생성
 - 대부분의 역할 수행 가능
 - 하지만 JSP는 서버 중심의 처리 방식으로 사용자가 많아지면 서버에 로드가 기하급수적으로 커짐
- 서버 중심의 처리 방식 문제점 해결을 위해 클라이언트 측에서 작업을 처리하는 다양한 기술 등장
 - 그 중 Ajax와 Json이 핵심 역할 수행

AJAX

- Asynchronous JavaScript and XML
 - JavaScript나 XML을 형식 데이터를 비동기식으로 전송하기 위한 기술
 - URL을 동일하게 유지하며, 내부적으로 여러개의 HTTP 요청과 응답을 전송할 수 있도록 지원
 - 웹 브라우저에서 페이지를 새로 고치지 않고도 여러개의 http 요청과 응답 가능



AJAX를 활용한 게시글 검색

▼ DB

```
CREATE TABLE `user` (  
  `userName` VARCHAR(50) NOT NULL COLLATE 'utf8mb4_0900_ai_ci',  
  `userAge` INT(10) NULL DEFAULT NULL,  
  `userGender` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',  
  `userEmail` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',  
  PRIMARY KEY (`userName`) USING BTREE  
)  
COLLATE='utf8mb4_0900_ai_ci'  
ENGINE=InnoDB  
;
```

▼ UserDTO.java

```

package jdbc;

public class UserDTO {
    private String userName;
    private int userAge;
    private String userGender;
    private String userEmail;
    public String getUserName() {
        return userName;
    }
    public void setUserName(String userName) {
        this.userName = userName;
    }
    public int getUserAge() {
        return userAge;
    }
    public void setUserAge(int userAge) {
        this.userAge = userAge;
    }
    public String getUserGender() {
        return userGender;
    }
    public void setUserGender(String userGender) {
        this.userGender = userGender;
    }
    public String getUserEmail() {
        return userEmail;
    }
    public void setUserEmail(String userEmail) {
        this.userEmail = userEmail;
    }
    public UserDTO(String userName, int userAge, String userGender, String userEmail) {
        super();
        this.userName = userName;
        this.userAge = userAge;
        this.userGender = userGender;
        this.userEmail = userEmail;
    }
}

```

▼ UserDao.java

```

package jdbc;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import org.json.simple.JSONArray;
import org.json.simple.JSONObject;

import util.ConnectionPool;

public class UserDao {

    // 게시물 목록 가져오기
    public List<UserDTO> selectList() {
        ArrayList<UserDTO> list = new ArrayList<>();
        String sql = "SELECT * FROM user";
        try {
            Connection conn = ConnectionPool.get();
            PreparedStatement pstmt = conn.prepareStatement(sql);

            ResultSet rSet = pstmt.executeQuery();

            while (rSet.next()) {
                UserDTO udto = new UserDTO(
                    rSet.getString("userName"),
                    rSet.getInt("userAge"),
                    rSet.getString("userGender"),
                    rSet.getString("userEmail")
                );
                list.add(udto);
            }

        } catch (SQLException e) {
            e.printStackTrace();
            list = null;
        }
    }
}

```

```

        return list;
    }

    // 게시물 검색
    public List<UserDTO> search(String userName) {
        String sql = "SELECT * FROM user WHERE userName LIKE ?";
        ArrayList<UserDTO> list = new ArrayList<>();
        Connection conn = null;
        PreparedStatement pstmt = null;
        ResultSet rSet = null;
        try {
            conn = ConnectionPool.get();
            pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, "%" + userName + "%");

            rSet = pstmt.executeQuery();

            while (rSet.next()) {
                UserDTO udto = new UserDTO(
                    rSet.getString("userName"),
                    rSet.getInt("userAge"),
                    rSet.getString("userGender"),
                    rSet.getString("userEmail")
                );
                list.add(udto);
            }
        } catch (SQLException e) {
            e.printStackTrace();
            list = null;
        } finally {
            try {
                rSet.close();
                pstmt.close();
                conn.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }

        return list;
    }
}

```

▼ board.jsp

```

<%@page import="jdbc.UserDTO"%>
<%@page import="java.util.List"%>
<%@page import="jdbc.UserDAO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    List<UserDTO> list = new UserDAO().selectList();
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Board</title>
</head>
<body>
<%@ include file="header.jsp" %>
<!-- nav bar -->
<nav class="navbar navbar-dark bg-dark">
    <div class="container-fluid">
        <a class="navbar-brand">BS</a>
        <input class="form-control me-2" id="userName" type="search" placeholder="Search" onkeyup="searchFunction()">
        <button class="btn btn-secondary btn-outline-dark" id="searchBtn" type="button" onclick="searchFunction();">Search</button>
    </div>
</nav>
<!-- Board Container -->
<div class="container">
    <table class="table table-hover">
        <thead>
            <tr>
                <th scope="col">name</th>
                <th scope="col">age</th>
                <th scope="col">gender</th>
                <th scope="col">email</th>
            </tr>
        </thead>
        <tbody id="table">
            <!-- Data rows would go here -->
        </tbody>
    </table>
</div>
<%@ include file="footer.jsp" %>

```

```

<script type="text/javascript">
var searchRequest = new XMLHttpRequest();
var registerRequest = new XMLHttpRequest();

function searchFunction() {
    searchRequest.open("Post", ". /UserSearchServlet?userName=" + encodeURIComponent(document.getElementById('userName').value), true);
    searchRequest.onreadystatechange = searchProcess;
    searchRequest.send(null);
}

function searchProcess() {
    var table = document.getElementById('table');
    table.innerHTML = "";
    if(searchRequest.readyState == 4 && searchRequest.status == 200) {
        var object = eval('(' + searchRequest.responseText + ')');
        var result = object.result;
        for(var i = 0; i < result.length; i++) {
            var row = table.insertRow(0);
            for(var j = 0; j < result[i].length; j++) {
                var cell = row.insertCell(j);
                cell.innerHTML = result[i][j].value;
            }
        }
    }
}
window.onload = function() {
    searchFunction();
}
</script>
</body>
</html>

```

▼ header.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQf"

```

▼ footer.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KFdBIe4zVF0s0G1M"
<!-- jquery -->
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXY0G+800xUf+/Im1MzjXxxg0cBQBXU=" crossorigin="anonymous">

```

JSON

JSON(JavaScript Object Notation)

JSON과 XML

- xml(Extensible Markup Language)

```

<dog>
  <name>식빵</name>
  <family>웰시코기</family>
  <age>1</age>
  <weight>2.14</weight>
</dog>

```

- Json

```

{
  "name": "식빵",
  "family": "웰시코기",
  "age": 1,
  "weight": 2.14
}

```

기존 xml형식의 데이터 처리보다 json이 쉬운 문법과 처리속도 측면에서도 우위에 있어 json을 더욱 많이 쓰는 추세임

DB데이터 JSON으로 받기

▼ DB

```
CREATE TABLE `user` (  
  `userName` VARCHAR(50) NOT NULL COLLATE 'utf8mb4_0900_ai_ci',  
  `userAge` INT(10) NULL DEFAULT NULL,  
  `userGender` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',  
  `userEmail` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',  
  PRIMARY KEY (`userName`) USING BTREE  
)  
COLLATE='utf8mb4_0900_ai_ci'  
ENGINE=InnoDB  
;
```

▼ UserDTO.java

```
package jdbc;  
  
public class UserDTO {  
    private String userName;  
    private int userAge;  
    private String userGender;  
    private String userEmail;  
    public String getUserName() {  
        return userName;  
    }  
    public void setUserName(String userName) {  
        this.userName = userName;  
    }  
    public int getUserAge() {  
        return userAge;  
    }  
    public void setUserAge(int userAge) {  
        this.userAge = userAge;  
    }  
    public String getUserGender() {  
        return userGender;  
    }  
    public void setUserGender(String userGender) {  
        this.userGender = userGender;  
    }  
    public String getUserEmail() {  
        return userEmail;  
    }  
    public void setUserEmail(String userEmail) {  
        this.userEmail = userEmail;  
    }  
    public UserDTO(String userName, int userAge, String userGender, String userEmail) {  
        super();  
        this.userName = userName;  
        this.userAge = userAge;  
        this.userGender = userGender;  
        this.userEmail = userEmail;  
    }  
}
```

▼ UserDAO.java

```
package jdbc;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.util.ArrayList;  
import java.util.List;  
  
import org.json.simple.JSONArray;  
import org.json.simple.JSONObject;  
  
import util.ConnectionPool;  
  
public class UserDAO {  
  
    // 게시물 검색 (JSON)  
    public String searchJson() {  
        String sql = "SELECT * FROM user";  
        JSONArray ja = new JSONArray();  
    }  
}
```

```

Connection conn = null;
PreparedStatement pstmt = null;
ResultSet rSet = null;
try {
    conn = ConnectionPool.get();
    pstmt = conn.prepareStatement(sql);
    rSet = pstmt.executeQuery();

    while (rSet.next()) {
        JSONObject jo = new JSONObject();
        jo.put("userName", rSet.getString("userName"));
        jo.put("userAge", rSet.getString("userAge"));
        jo.put("userGender", rSet.getString("userGender"));
        jo.put("userEmail", rSet.getString("userEmail"));
        ja.add(jo);
    }

} catch (SQLException e) {
    e.printStackTrace();
} finally {
    try {
        rSet.close();
        pstmt.close();
        conn.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

return ja.toJSONString();
}
}

```

▼ jsonList.jsp

```

<%@page import="jdbc.UserDAO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%
    out.print(new UserDAO().searchJson());
%>

```

응용 - 게시판 로드없이 바로 등록

▼ DB

```

CREATE TABLE `user` (
  `userName` VARCHAR(50) NOT NULL COLLATE 'utf8mb4_0900_ai_ci',
  `userAge` INT(10) NULL DEFAULT NULL,
  `userGender` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',
  `userEmail` VARCHAR(50) NULL DEFAULT NULL COLLATE 'utf8mb4_0900_ai_ci',
  PRIMARY KEY (`userName`) USING BTREE
)
COLLATE='utf8mb4_0900_ai_ci'
ENGINE=InnoDB
;

```

▼ UserDTO.java

```

package jdbc;

public class UserDTO {
    private String userName;
    private int userAge;
    private String userGender;
    private String userEmail;
    public String getUserName() {
        return userName;
    }
}

```

```

    public void setUserName(String userName) {
        this.userName = userName;
    }
    public int getUserAge() {
        return userAge;
    }
    public void setUserAge(int userAge) {
        this.userAge = userAge;
    }
    public String getUserGender() {
        return userGender;
    }
    public void setUserGender(String userGender) {
        this.userGender = userGender;
    }
    public String getUserEmail() {
        return userEmail;
    }
    public void setUserEmail(String userEmail) {
        this.userEmail = userEmail;
    }
}
public UserDTO(String userName, int userAge, String userGender, String userEmail) {
    super();
    this.userName = userName;
    this.userAge = userAge;
    this.userGender = userGender;
    this.userEmail = userEmail;
}
}
}

```

▼ UserDAO.java

```

package jdbc;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;

import org.json.simple.JSONArray;
import org.json.simple.JSONObject;

import util.ConnectionPool;

public class UserDAO {
    // 회원 등록
    public int register(String userName, int userAge, String userGender, String userEmail) {
        int result = 0;
        String sql = "INSERT INTO user (userName, userAge, userGender, userEmail) VALUES (?, ?, ?, ?)";

        try {
            Connection conn = ConnectionPool.get();
            PreparedStatement pstmt = conn.prepareStatement(sql);
            pstmt.setString(1, userName);
            pstmt.setInt(2, userAge);
            pstmt.setString(3, userGender);
            pstmt.setString(4, userEmail);

            result = pstmt.executeUpdate();
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }

        return result;
    }

    // 회원 목록 가져오기
    public List<UserDTO> selectList() {
        ArrayList<UserDTO> list = new ArrayList<>();
        String sql = "SELECT * FROM user";
        try {
            Connection conn = ConnectionPool.get();
            PreparedStatement pstmt = conn.prepareStatement(sql);

            ResultSet rSet = pstmt.executeQuery();

            while (rSet.next()) {
                UserDTO udto = new UserDTO(
                    rSet.getString("userName"),
                    rSet.getInt("userAge"),

```

```

        rSet.getString("userGender"),
        rSet.getString("userEmail")
    );
    list.add(udto);
}

} catch (SQLException e) {
    e.printStackTrace();
    list = null;
}

return list;
}

// 회원 검색
public List<UserDTO> search(String userName) {
    String sql = "SELECT * FROM user WHERE userName LIKE ?";
    ArrayList<UserDTO> list = new ArrayList<>();
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rSet = null;
    try {
        conn = ConnectionPool.get();
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, "%" + userName + "%");

        rSet = pstmt.executeQuery();

        while (rSet.next()) {
            UserDTO udto = new UserDTO(
                rSet.getString("userName"),
                rSet.getInt("userAge"),
                rSet.getString("userGender"),
                rSet.getString("userEmail")
            );
            list.add(udto);
        }
    } catch (SQLException e) {
        e.printStackTrace();
        list = null;
    } finally {
        try {
            rSet.close();
            pstmt.close();
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    return list;
}

```

▼ UserRegisterServlet.java

```

package jdbc;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

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

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

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        response.setContentType("text/html;charset=UTF-8");
        String userName = request.getParameter("userName");
        String userAge = request.getParameter("userAge");
        String userGender = request.getParameter("userGender");
        String userEmail = request.getParameter("userEmail");
        response.getWriter().write(register(userName, userAge, userGender, userEmail) + "");
    }

    public int register(String userName, String userAge, String userGender, String userEmail) {
        return new UserDao().register(userName, Integer.parseInt(userAge), userGender, userEmail);
    }
}

```



```
}
```

▼ UserSearchServlet.java

```
package jdbc;

import java.io.IOException;
import java.util.ArrayList;
import java.util.List;

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

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

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("UTF-8");
        response.setContentType("text/html;charset=UTF-8");
        String userName = request.getParameter("userName");
        response.getWriter().write(getJSON(userName));
    }

    public String getJSON(String userName) {
        if(userName == null) userName = "";
        StringBuffer result = new StringBuffer("");
        result.append("{\"result\":[");
        UserDao udao = new UserDao();
        List<UserDTO> userList = udao.search(userName);
        for(int i = 0; i < userList.size(); i++) {
            result.append("[{\"value\": \"" + userList.get(i).getUserName() + "\",");
            result.append("{\"value\": \"" + userList.get(i).getUserAge() + "\",");
            result.append("{\"value\": \"" + userList.get(i).getUserGender() + "\",");
            result.append("{\"value\": \"" + userList.get(i).getUserEmail() + "\"}],");
        }
        result.append("]");
        return result.toString();
    }
}
```

▼ board.jsp

```
<%@page import="jdbc.UserDTO"%>
<%@page import="java.util.List"%>
<%@page import="jdbc.UserDAO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>

<%
    List<UserDTO> list = new UserDao().selectList();
%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Board</title>
</head>
<body>
<%@ include file="header.jsp" %>
<!-- nav bar -->
<nav class="navbar navbar-dark bg-dark">
    <div class="container-fluid">
        <a class="navbar-brand">BS</a>
        <input class="form-control me-2" id="userName" type="search" placeholder="Search" onkeyup="searchFunction()">
        <button class="btn btn-secondary btn-outline-dark" id="searchBtn" type="button" onclick="searchFunction();">Search</button>
    </div>
</nav>
<!-- Board Container -->
<div class="container">
    <table class="table table-hover">
        <thead>
            <tr>
                <th scope="col">name</th>
                <th scope="col">age</th>
                <th scope="col">gender</th>
                <th scope="col">email</th>
            </tr>
        </thead>
    </table>
</div>
```

```

        </tr>
    </thead>
    <tbody id="table">
    </tbody>
    </table>
</div>
<!-- Board Container -->
<div class="container mt-5">
    <form class="row g-3">
        <div class="col-12">
            <label for="inputName" class="form-label">Name</label>
            <input type="text" class="form-control" id="registerName" name="registerName">
        </div>
        <div class="col-12">
            <label for="inputAge" class="form-label">Age</label>
            <input type="text" class="form-control" id="registerAge" name="registerAge">
        </div>
        <div class="col-12">
            <label class="form-label">Gender</label>
            <div class="form-check">
                <input class="" type="radio" name="registerGender" value="male" checked>
                <label class="form-check-label" for="male">
                    male
                </label>
                <input class="" type="radio" name="registerGender" value="female">
                <label class="form-check-label" for="female">
                    female
                </label>
            </div>
        </div>
        <div class="col-12">
            <label for="inputEmail4" class="form-label">Email</label>
            <input type="email" class="form-control" id="registerEmail" name="registerEmail">
        </div>
        <div class="col-12">
            <button type="button" class="btn btn-primary" onclick="registerFunction()">Sign up</button>
        </div>
    </form>
</div>
<%% include file="footer.jsp" %>
<script type="text/javascript">
    var searchRequest = new XMLHttpRequest();
    var registerRequest = new XMLHttpRequest();

    function searchFunction() {
        searchRequest.open("Post", ". /UserSearchServlet?userName=" + encodeURIComponent(document.getElementById('userName').value), true);
        searchRequest.onreadystatechange = searchProcess;
        searchRequest.send(null);
    }

    function searchProcess() {
        var table = document.getElementById('table');
        table.innerHTML = "";
        if(searchRequest.readyState == 4 && searchRequest.status == 200) {
            var object = eval('(' + searchRequest.responseText + ')');
            var result = object.result;
            for(var i = 0; i < result.length; i++) {
                var row = table.insertRow(0);
                for(var j = 0; j < result[i].length; j++) {
                    var cell = row.insertCell(j);
                    cell.innerHTML = result[i][j].value;
                }
            }
        }
    }

    function registerFunction() {
        registerRequest.open("Post", ". /UserRegisterServlet?userName=" + encodeURIComponent(document.getElementById('registerName').value)
        + "&userAge=" + encodeURIComponent(document.getElementById('registerAge').value)
        + "&userGender=" + encodeURIComponent($('input[name=registerGender]:checked').val())
        + "&userEmail=" + encodeURIComponent(document.getElementById('registerEmail').value)
        , true);
        registerRequest.onreadystatechange = registerProcess;
        registerRequest.send(null);
    }

    function registerProcess() {
        if(registerRequest.readyState == 4 && registerRequest.status == 200) {
            var result = registerRequest.responseText;
            if(result != 1) {
                alert('등록에 실패했습니다.');
```

```

        registerEmail.value = "";
        searchFunction();
    }
}
}
window.onload = function() {
    searchFunction();
}
</script>
</body>
</html>

```

▼ header.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQf"

```

▼ footer.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!-- JavaScript Bundle with Popper -->
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KfDBIe4zVF0s0G1P"
<!-- jquery -->
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXY0G+800xUf+/Im1MZjXxxgOcBQBxU=" crossorigin="no"

```