AJAX & JSON

✔️***AJAX***

✔️***Ajax basic***

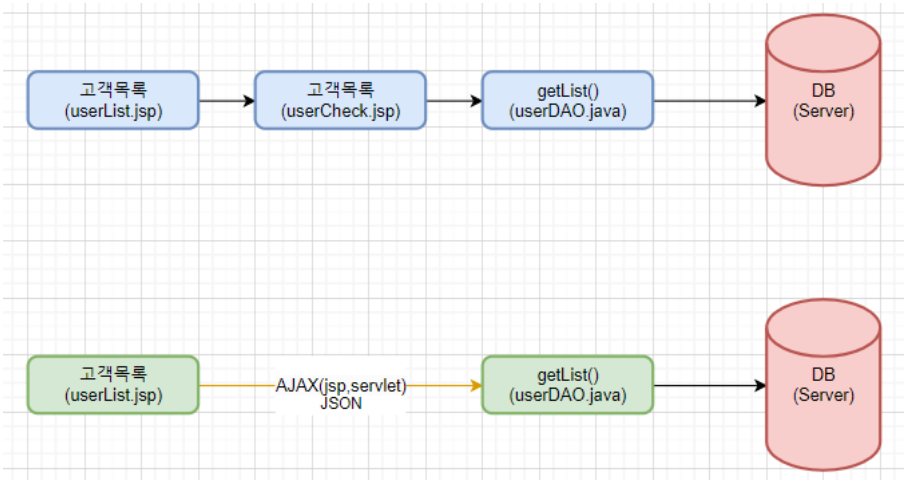
✔️***Ajax Adv***

✔️***JSON***

AJAX

JSP 는 사용자의 요청에 대한 결과 페이지를 생성하기 위해 대부분의 역할 수행한다. 이런 방식은 서버 중심의 처리 방식으로 볼 수 있다. 사용자가 많아지게 되면 서버에 로드가 기하급수적으로 커지는 문제점이 있다.

이러한 문제점은 페이스북, 트위터 등의 SNS가 등장하면서 더욱 현실화 되었다. 이를테면 SNS 서버에는 동시 사용자가 10만명 30만명 등등 이 넘어가는 경우가 비일비재하다. 따라서 서버의 부하를 줄이기 위해 서버가 하던 작업을 클라이언트로 넘기는 다양한 기술들이 등장했다. 그중 AJAX와 JSON 이 핵심적인 역할을 수행하게 된다.



AJAX '**Asynchronous** JAVA and XML' JAVA나 XML형식의 데이터를 **비동기식**으로 전송하기 위한 기술

AJAX는 URL을 동일하게 유지하면서 내부적으로 여러개의 HTTP 요청과 응답을 전송할수 있도록 지원한다. 이를 통해 웹 브라우저에서 페이지를 고치지 않고도 여러개의 http 요청과 응답을 가능하게 한다.

Ajax basic

user Table

Column	Type
userName	varchar(50)
userAge	int
userGender	varchar(50)
userEmail	varchar(50)

userList.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"
    import="user.*, java.util.*"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>userList</title>
</head>
<body>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLhlTQ8
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXY0G+8O0xUf+/Im1MZjXxxgOcBQBXU=" crossorigin=

<script type="text/javascript">
  var searchRequest = 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('ajaxTable');
    table.innerHTML = "<tr>";
    if(searchRequest.readyState == 4 && searchRequest.status == 200){
      var object = eval('(' + searchRequest.responseText + ')');
      var result = object.result;
      for(var i=0; i<result.length; i++){//유저 데이터 한명의 자료 추출
        var row = table.insertRow(0);
        for(var j=0; j<result[i].length; j++){//한명의 필드 추출
          var cell = row.insertCell(j);
        cell.innerHTML = result[i][j].value;
        }
      }
    }
  }
</script>

<div class="container">
<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="nav
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active" aria-current="page" href="#">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#">Link</a>
        </li>
        <li class="nav-item dropdown">
          <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">
            Dropdown
          </a>
          <ul class="dropdown-menu">
            <li><a class="dropdown-item" href="#">Action</a></li>
            <li><a class="dropdown-item" href="#">Another action</a></li>
            <li><hr class="dropdown-divider"></li>
            <li><a class="dropdown-item" href="#">Something else here</a></li>
          </ul>
        </li>
        <li class="nav-item">
          <a class="nav-link disabled">Disabled</a>
        </li>
      </ul>
      <input class="form-control me-2" type="search" id="userName" onkeyup="searchFunction()" placeholder="Search" aria-label="Search"
        <button class="btn btn-outline-primary" type="submit" onclick="searchFunction();">Search</button>
    </div>
  </div>
</nav>
</div>

<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="ajaxTable">
    </tbody>
</table>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-w76AqPfDkMBDXo30jS
</body>
</html>
```

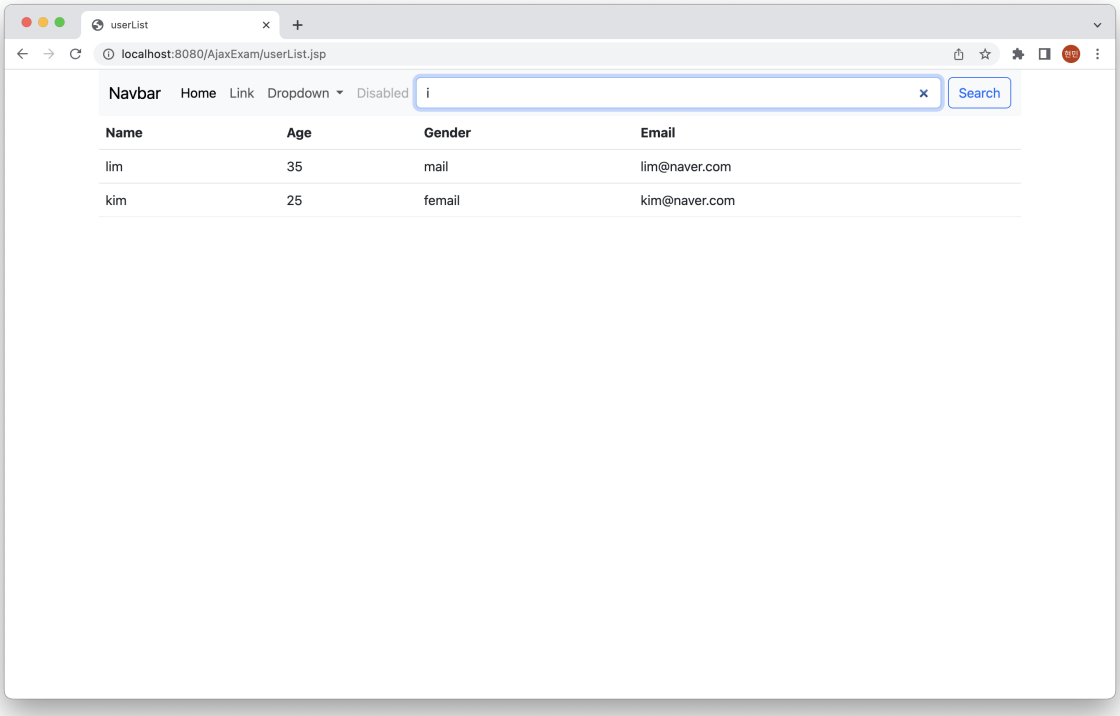
UserDAO.java

```
package user;

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

import javax.naming.NamingException;

import user.User;
import util.ConnectionPool;

public class UserDAO {
  public static ArrayList<User> search(String userName) {
    String sql = "SELECT * FROM user WHERE userName LIKE ?";

    ArrayList<User> users = new ArrayList<User>();

    try {
      Connection conn = ConnectionPool.get();

      PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, '%'+userName+'%');

      ResultSet rs = pstmt.executeQuery();

      while(rs.next()) {
        users.add(new User(rs.getString(1), rs.getInt(2), rs.getString(3), rs.getString(4)));
      }
    } catch (Exception e) {
      e.printStackTrace();
    }

    return users;
  }
}
```

UserSearchServlet.java

```
package user;

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

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 userDAO = new UserDAO();
    ArrayList<User> userList = userDAO.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();
  }

}
```



Ajax Adv

json

JavaScript Object Notation,

- 자바스크립트에서 객체를 표현하기 위한 형식
- XML과 아주 유사하지만 xml에 비해 쉬운 문법을 사용하고 처리속도도 빠르다는 장점이 있다

따라서 모바일 앱 등의 구현에 있어서 json이 점점더 많이 사용되고 있다.

자바 스크립트에서는 객체를 중괄호로 정의한다. 객체는 이름-값 의 쌍 형태로 정의된 속성을 하나 이상 포함 할수 있고 각각의 속성은 쉼표로 구분된다. 이때 이름은 스트링 형식으로 표현되고 값은 임의 자료형으로 정의 될수 있다.

💡 {

id:"Kim@naver.com",

pass:"0000",

name:"kim"

}

배열 형식으로 표현할 수 있다.

{

0:"Kim@naver.com",

1:"0000",

2:"kim"

}

배열 형태

{"Kim@naver.com", "0000", "kim"}

[

{id:"Kim@naver.com",pass:"0000",name:"kim"},

{id:"hong@yahoo.com",pass:"9876",name:"hong"},

{id:"yang@daum.com",pass:"3456",name:"yang"}

]

이러한 json 배열을 클라이언트로 전송하여 html 로 출력하게 된다.

UserDAO getJSON method

```
public static String getListJSON() throws NamingException, SQLException{
    String sql = "SELECT * FROM user";

    Connection conn = ConnectionPool.get();
```

```
    PreparedStatement pstmt = conn.prepareStatement(sql);

    ResultSet rs = pstmt.executeQuery();

    JSONArray users = new JSONArray();

    while(rs.next()) {
      JSONObject obj = new JSONObject();
      obj.put("userName", rs.getString("userName"));
      obj.put("userAge", rs.getString("userAge"));
      obj.put("userGender", rs.getString("userGender"));
      obj.put("userEmail", rs.getString("userEmail"));
      users.add(obj);
    }

    return users.toJSONString();
  }
```

JsonList.jsp

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