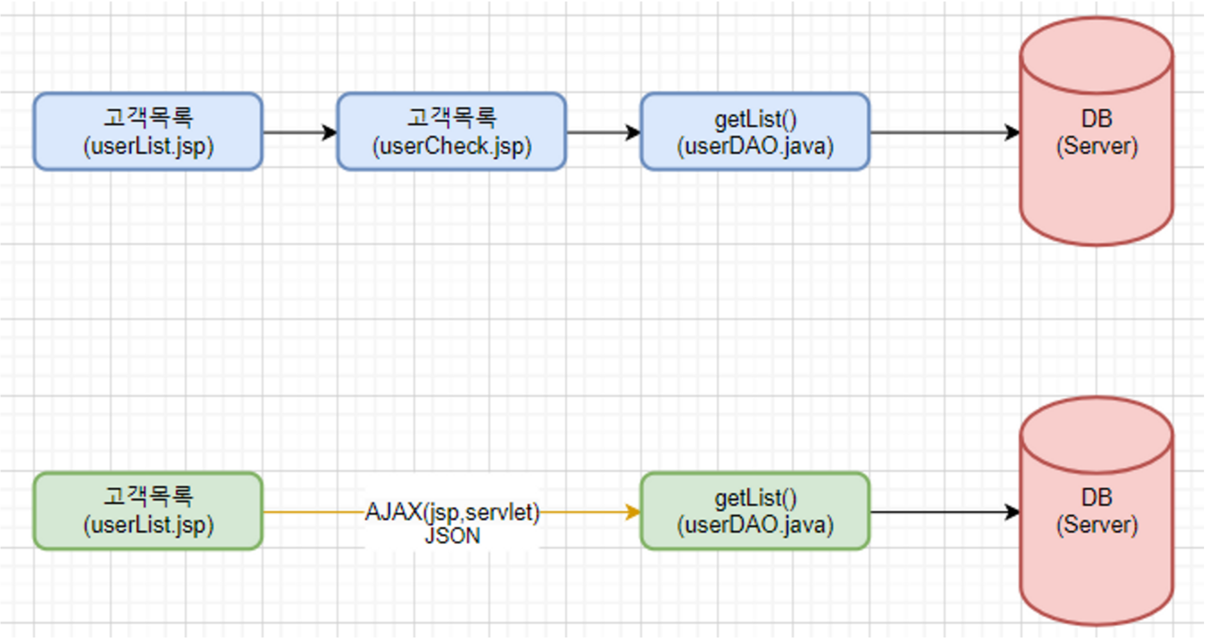
20230127 AJAX & JSON

날짜	@2023년 1월 27일
태그	

AJAX

JSP는 사용자의 요청에 대한 결과 페이지를 생성하기 위해 대부분의 역학을 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 JSON이 핵심적인 역할을 수행하게 된다.



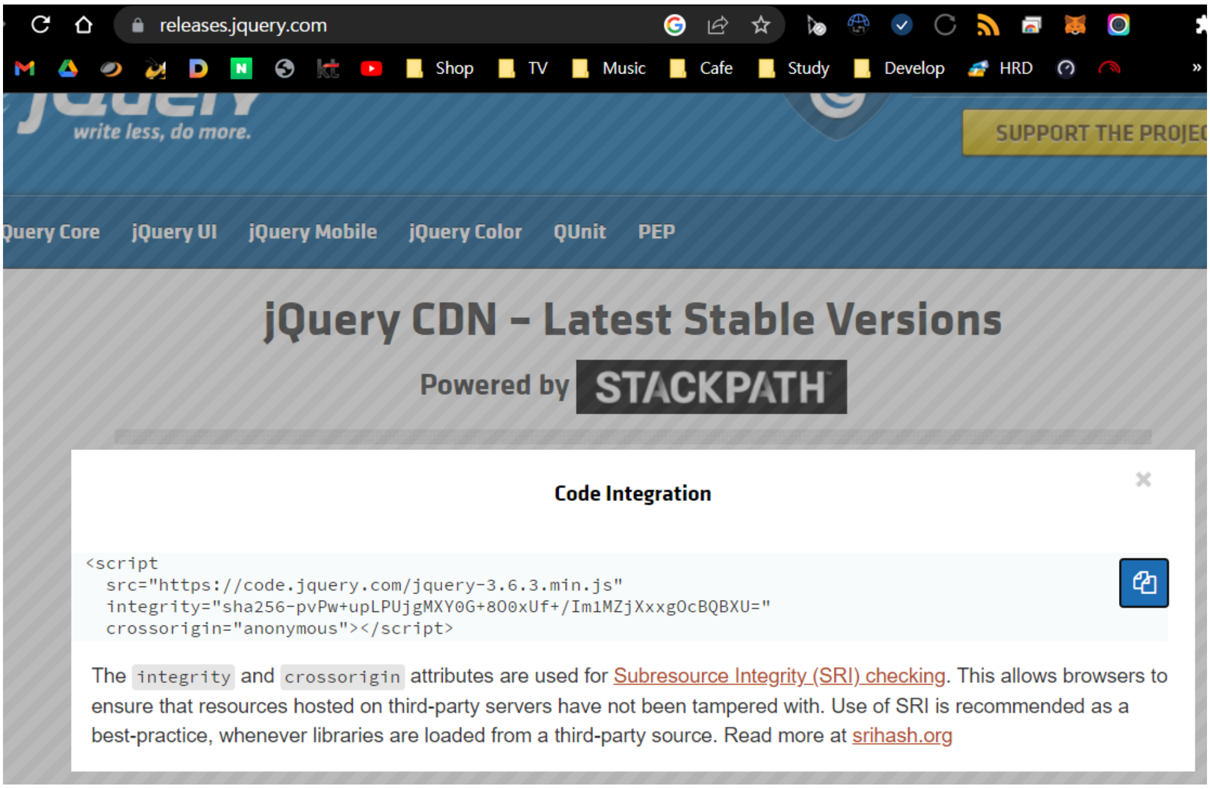
- AJAX (Asynchronous JAVA and XML)

  JAVA나 XML형식의 데이터를 **비동기식**으로 전송하기 위한 기술

  AJAX는 URL을 동일하게 유지하면서 내부적으로 여러개의 HTTP 요청과 응답을 전송할수 있도록 지원한다.

  웹 브라우저에서 페이지를 고치지 않고도 여러개의 http 요청과 응답을 가능하게 한다.

jquery 설정



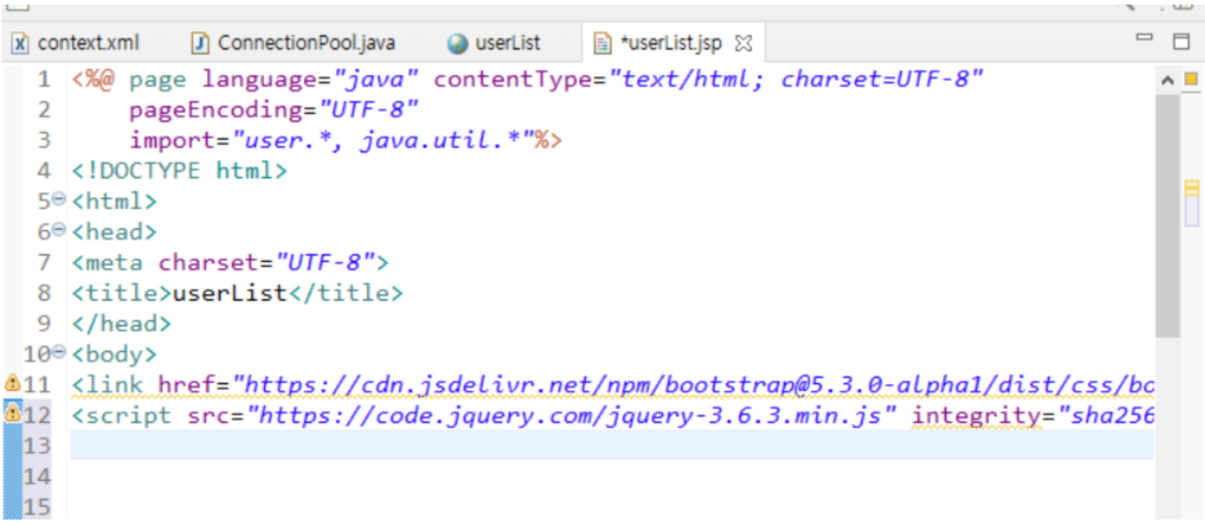
jQuery CDN – Latest Stable Versions

Powered by STACKPATH

Code Integration

```
<script
  src="https://code.jquery.com/jquery-3.6.3.min.js"
  integrity="sha256-pvPw+upLPUjgMXY0G+8O0xUf+/Im1MZjXxxgOcBQBXU="
  crossorigin="anonymous"></script>
```

The `integrity` and `crossorigin` attributes are used for Subresource Integrity (SRI) checking. This allows browsers to ensure that resources hosted on third-party servers have not been tampered with. Use of SRI is recommended as a best-practice, whenever libraries are loaded from a third-party source. Read more at srihash.org



```
<%@ 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/bc
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256
```

```
<form class="d-flex" role="search">
<input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
<button class="btn btn-outline-primary" type="submit">Search</button>
</form>
```

기존에 사용하던 form 태그는 더 이상 사용되지 않는다. form 태그는 필연적으로 처리 페이지 로의 전달을 위해서 화면 전환이 이루어지게 된다. 따라서 jqeury 사용 시 form 태그를 전혀 사용하지 않는다

form 태그 대신

onkeyup="searchFunction()" 키 입력시 마다 함수 호출

onclick="searchFunction();" 단추를 누를 경우 함수 호출 방식으로

처리가 변경된다.

▼ ajax basic

- main.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>JSP AJAX</title>

</head>
<body>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLh
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXY0G+8O0xUf+/Im1MZjXxxgOcBQBXU=" crossorig

<script type="text/javascript">
  var searchRequest = new XMLHttpRequest();
  var registerRequest = new XMLHttpRequest();
  function searchFunction() {
    searchRequest.open("Post", "./UserSearchServlet?userName=" + encodeURIComponent(document.getElementById('userName').value), tru
    searchRequest.onreadystatechange = searchProcess;
    searchRequest.send(null);
  }
  function searchProcess() {
    var table = document.getElementById('ajaxTable');
    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;
        }
      }
    }
  }

</script>


<br>
<nav class="navbar bg-dark" data-bs-theme="dark">
  <div class="container-fluid">
    <a class="navbar-brand">Navbar</a>
    <form class="d-flex" role="search">
          <input class="form-control me-2" type="search" id="userName" onkeyup="searchFunction()" placeholder="Search" aria-label="
          <button class="btn btn-outline-success" onclick="searchFunction();" type="button">Search</button>

    </form>
</nav>


<br>


<div class="container-sm">

  <table class="table table-dark 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>
<br>
<br>
<!-- <div class="container">
  <table class="table table-dark">
    <thead>
      <tr>
        <th colspan="2" style="text-align: center;">Register</th>
      </tr>
    </thead>
    <tbody>
```

```
      <tr>
        <td style="text-align: center;">Name</td>
        <td><input class="form-control" type="text" id="registerName"></td>
      </tr>
      <tr>
        <td style="text-align: center;">Age</td>
        <td><input class="form-control" type="text" id="registerAge"></td>
      </tr>
      <tr>
        <td style="text-align: center;">Gender</td>
        <td>
          <div class="form-group" style="text-align: center;">
            <div class="btn-group" data-toggle="buttons">
              <label class="btn btn-success">
                <input type="radio" name="registerGender" value="male">Male
              </label>
              <label class="btn btn-danger">
                <input type="radio" name="registerGender" value="female" checked>Female
              </label>
            </div>
          </div>
        </td>
      </tr>
      <tr>
        <td style="text-align: center;">Email</td>
        <td><input class="form-control" type="text" id="registerEmail" size="20"></td>
      </tr>
      <tr>
        <td colspan="2" style="text-align: center;"><button class="btn btn-primary pull-right" onclick="registerFunction();" type="
      </tr>
    </tbody>
  </table>
</div>
 -->

</body>
</html>
```

- UserSearchServlet

```
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();
  }

}
```

- userDAO

```
public ArrayList<User> search(String userName) {
    String SQL = "SELECT * FROM USER WHERE userName LIKE ?";
    ArrayList<User> userList = new ArrayList<User>();
    try {
      pstmt = conn.prepareStatement(SQL);
      pstmt.setString(1, "%" + userName + "%");
      rs = pstmt.executeQuery();
      while (rs.next()) {
        User user = new User();
        user.setUserName(rs.getString(1));
        user.setUserAge(rs.getInt(2));
        user.setUserGender(rs.getString(3));
        user.setUserEmail(rs.getString(4));
        userList.add(user);
      }
    } catch(Exception e) {
      e.printStackTrace();
    }
    return userList;
  }
```

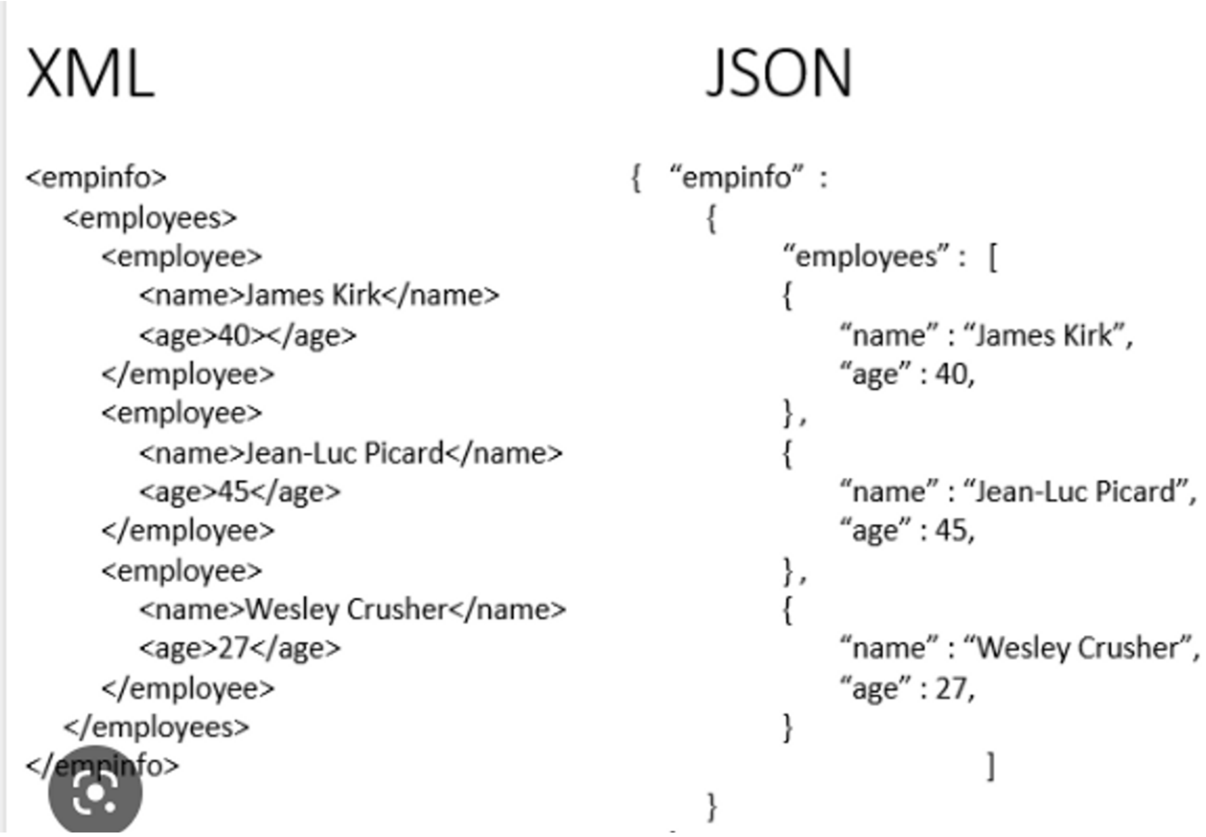
ajax adv

json

JavaScript Object Notation (JSON)

자바스크립트에서 객체를 표현하기 위한 형식

xml과 아주 유사하지만 xml에 비해 쉬운 문법을 사용하고 처리속도도 빠르다는 장점이 있다.



XML

```
<empinfo>
   <employees>
      <employee>
         <name>James Kirk</name>
         <age>40></age>
      </employee>
      <employee>
         <name>Jean-Luc Picard</name>
         <age>45</age>
      </employee>
      <employee>
         <name>Wesley Crusher</name>
         <age>27</age>
      </employee>
   </employees>
</empinfo>
```

JSON

```
{  “empinfo” :
      {
            “employees” :  [
            {
                “name” : “James Kirk”,
                “age” : 40,
            },
            {
                “name” : “Jean-Luc Picard”,
                “age” : 45,
            },
            {
                “name” : “Wesley Crusher”,
                “age” : 27,
            }
                           ]
      }
```

따라서 모바일 앱 등의 구현에 있어서 json이 더 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.

- 객체…

```
{

  id:”Kim@naver.com”,

  pass:”0000”,

  name:”kim”

}
```

- 배열 형식으로도 표현할 수 있다

```
{

  0:”Kim@naver.com”,

  1:”0000”,

  2:”kim”

}
```

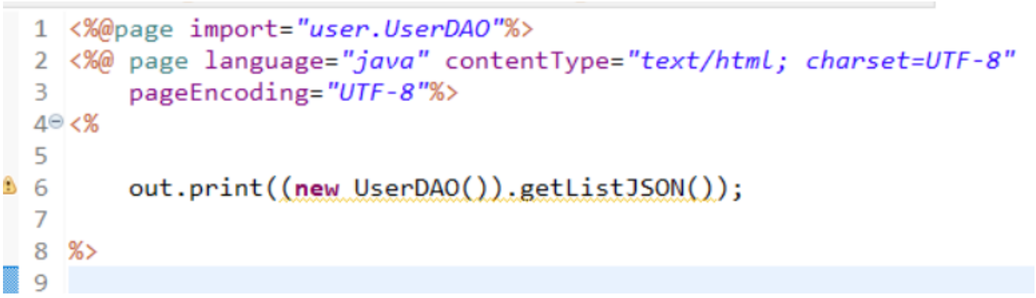
- 배열 형태

```
{”Kim@naver.com”, ”0000”, ”kim”}
```

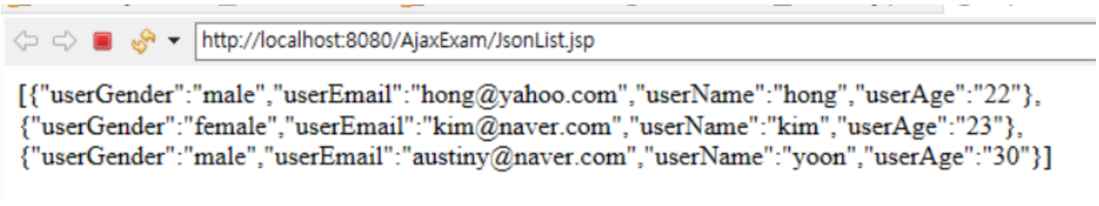
```
[

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

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

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

]
```

이러한 JSON 배열을 클라이언트로 전송하여 html로 출력

```
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();
  }
```



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

    out.print((new UserDAO()).getListJSON());

%>

```



```
[{"userGender":"male","userEmail":"hong@yahoo.com","userName":"hong","userAge":"22"},
{"userGender":"female","userEmail":"kim@naver.com","userName":"kim","userAge":"23"},
{"userGender":"male","userEmail":"austiny@naver.com","userName":"yoon","userAge":"30"}]
```

최종 버전

▼ main.jsp

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<title>JSP AJAX</title>

</head>
<body>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha38
<script src="https://code.jquery.com/jquery-3.6.3.min.js" integrity="sha256-pvPw+upLPUjgMXY0G+8O0xUf+/Im1MZjXxxgOcBQBXU=" cro

<script type="text/javascript">
  var searchRequest = new XMLHttpRequest();
  var registerRequest = new XMLHttpRequest();
  function searchFunction() {
    searchRequest.open("Post", "./UserSearchServlet?userName=" + encodeURIComponent(document.getElementById('userName').value
    searchRequest.onreadystatechange = searchProcess;
    searchRequest.send(null);
  }
  function searchProcess() {
    var table = document.getElementById('ajaxTable');
    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
      + "&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.');
      } else {
        var userName = document.getElementById('userName');
        var registerName = document.getElementById('registerName');
        var registerAge = document.getElementById('registerAge');
        var registerEmail = document.getElementById('registerEmail');
        userName.value = "";
        registerName.value = "";
        registerAge.value = "";
        registerEmail.value = "";
        searchFunction();
      }
    }
  }


  window.onload = function() {
    searchFunction();
  }
</script>


<br>
<nav class="navbar bg-dark" data-bs-theme="dark">
  <div class="container-fluid">
    <a class="navbar-brand">Navbar</a>
    <form class="d-flex" role="search">
          <input class="form-control me-2" type="search" id="userName" onkeyup="searchFunction()" placeholder="Search" aria-la
          <button class="btn btn-outline-success" onclick="searchFunction();" type="button">Search</button>

    </form>
</nav>


<br>


<div class="container-sm">

  <table class="table table-dark 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>
<br>
<br>
<div class="container">
  <table class="table table-dark">
    <thead>
      <tr>
        <th colspan="2" style="text-align: center;">Register</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td style="text-align: center;">Name</td>
        <td><input class="form-control" type="text" id="registerName"></td>
      </tr>
      <tr>
        <td style="text-align: center;">Age</td>
        <td><input class="form-control" type="text" id="registerAge"></td>
      </tr>
      <tr>
        <td style="text-align: center;">Gender</td>
        <td>
          <div class="form-group" style="text-align: center;">
            <div class="btn-group" data-toggle="buttons">
              <label class="btn btn-success">
                <input type="radio" name="registerGender" value="male">Male
              </label>
              <label class="btn btn-danger">
                <input type="radio" name="registerGender" value="female" checked>Female
              </label>
            </div>
          </div>
        </td>
      </tr>
      <tr>
        <td style="text-align: center;">Email</td>
        <td><input class="form-control" type="text" id="registerEmail" size="20"></td>
      </tr>
      <tr>
        <td colspan="2" style="text-align: center;"><button class="btn btn-primary pull-right" onclick="registerFunction();" 
      </tr>
    </tbody>
  </table>
</div>


</body>
</html>
```

▼ UserRegisterServlet

```
package user;

import java.io.IOException;
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) {
    User user = new User();
    try {
      user.setUserName(userName);
      user.setUserAge(Integer.parseInt(userAge));
      user.setUserGender(userGender);
      user.setUserEmail(userEmail);
    } catch(Exception e) {
      return 0;
    }
    return new UserDAO().register(user);
  }
}
```

▼ UserDAO (DB 설정 부분이 ConnenrionPool 이 아닌 다른 방식)

```
package user;

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

public class UserDAO {
  private Connection conn;
  private PreparedStatement pstmt;
  private ResultSet rs;

  public UserDAO() {
    try {
      String dbURL = "jdbc:mysql://localhost:3306/market";
      String dbID = "root";
      String dbPassword = "0000";
      Class.forName("com.mysql.jdbc.Driver");
      conn = DriverManager.getConnection(dbURL, dbID, dbPassword);
    } catch(Exception e) {
      e.printStackTrace();
    }
  }
  public ArrayList<User> search(String userName) {
    String SQL = "SELECT * FROM USER WHERE userName LIKE ?";
    ArrayList<User> userList = new ArrayList<User>();
    try {
      pstmt = conn.prepareStatement(SQL);
      pstmt.setString(1, "%" + userName + "%");
      rs = pstmt.executeQuery();
      while (rs.next()) {
        User user = new User();
        user.setUserName(rs.getString(1));
        user.setUserAge(rs.getInt(2));
        user.setUserGender(rs.getString(3));
        user.setUserEmail(rs.getString(4));
        userList.add(user);
      }
    } catch(Exception e) {
      e.printStackTrace();
    }
    return userList;
  }

  public int register(User user) {
    String SQL = "INSERT INTO  USER VALUES (?, ?, ?, ?)";
    try {
      pstmt = conn.prepareStatement(SQL);
      pstmt.setString(1, user.getUserName());
      pstmt.setInt(2, user.getUserAge());
      pstmt.setString(3, user.getUserGender());
      pstmt.setString(4, user.getUserEmail());
      return pstmt.executeUpdate(); // return 1 (행의 수)
    } catch(Exception e) {
      e.printStackTrace();
    }
    return -1; // 데이터베이스 오류
  }
}
```