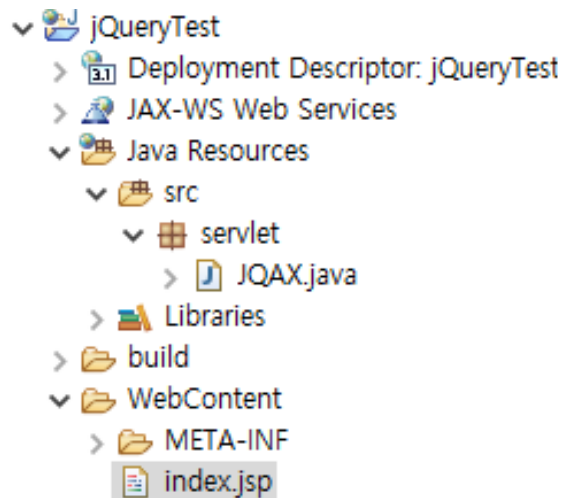


# 통합 function 사용



```
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2   pageEncoding="utf-8"%>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="utf-8">
7     <title>jQuery AJAX</title>
8     <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
9   </head>
10  <body>
11    <p>통신요청</p>
12    <script>
13      //$.ajax(url);
14      //$.ajax(url, option);
15      $("p").click(function(){
16        $.ajax("JQAX");
17      });
18    </script>
19  </body>
20 </html>
```

```
1 package servlet;
2
3 import java.io.*;
4 import javax.servlet.*;
5 import javax.servlet.annotation.*;
6 import javax.servlet.http.*;
7
8 @WebServlet("/JQAX")
9 public class JQAX extends HttpServlet {
10   private static final long serialVersionUID = 1L;
11
12   protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
13     System.out.println("통신요청 확인");
14   }
15
16   protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
17   }
18 }
19 }
```

**\$.ajax를 이용하면 통합 요청이 가능하다.**

# success 값 이용

```
1 package servlet;
2
3 import java.io.*;
4 import javax.servlet.*;
5 import javax.servlet.annotation.*;
6 import javax.servlet.http.*;
7
8 @WebServlet("/JQAX")
9 public class JQAX extends HttpServlet {
10     private static final long serialVersionUID = 1L;
11
12     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
13         System.out.println("통신요청 확인");
14         response.setCharacterEncoding("utf-8");
15         response.getWriter().write("다음값을 반환");
16     }
17
18     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
19     }
20 }
21 }
```

```
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2     pageEncoding="utf-8"%>
3 <!DOCTYPE html>
4 <html>
5     <head>
6         <meta charset="utf-8">
7         <title>jQuery AJAX</title>
8         <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
9     </head>
10    <body>
11        <p>통신요청</p>
12        <h1 id="response"></h1>
13        <h1 id="status"></h1>
14        <h1 id="xhr"></h1>
15        <script>
16            //$.ajax(url);
17            //$.ajax(url, option);
18            $("p").click(function(){
19                $.ajax("JQAX", {
20                    success : function(response, status, xhr){
21                        $("#response").text(response);
22                        $("#status").text(status);
23                        $("#xhr").text(xhr);
24                    }
25                });
26            });
27        </script>
28    </body>
29 </html>
```

통신 성공시 반환값을 사용할 수 있다.

# 추가 속성값의 사용

```
1 <%@ page language="java" contentType="text/html; charset=utf-8"
2   pageEncoding="utf-8"%>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="utf-8">
7     <title>jQuery AJAX</title>
8     <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
9   </head>
10  <body>
11    <p>통신요청</p>
12    <h1 id="response"></h1>
13    <h1 id="status"></h1>
14    <h1 id="xhr"></h1>
15    <script>
16      //$.ajax(url);
17      //$.ajax(url, option);
18      $("p").click(function(){
19        $.ajax("JQAX", {
20          type : "POST",
21          data : {
22            name: "폴이",
23            race: "말티즈"
24          },
25          success : function(response, status, xhr){
26            $("#response").text(response);
27            $("#status").text(status);
28            $("#xhr").text(xhr);
29          },
30          error : function(xhr, status, errorMessage){
31            $("#response").text(xhr);
32            $("#status").text(status);
33            $("#xhr").html(errorMessage);
34          }
35        });
36      });
37    </script>
38  </body>
39 </html>
```

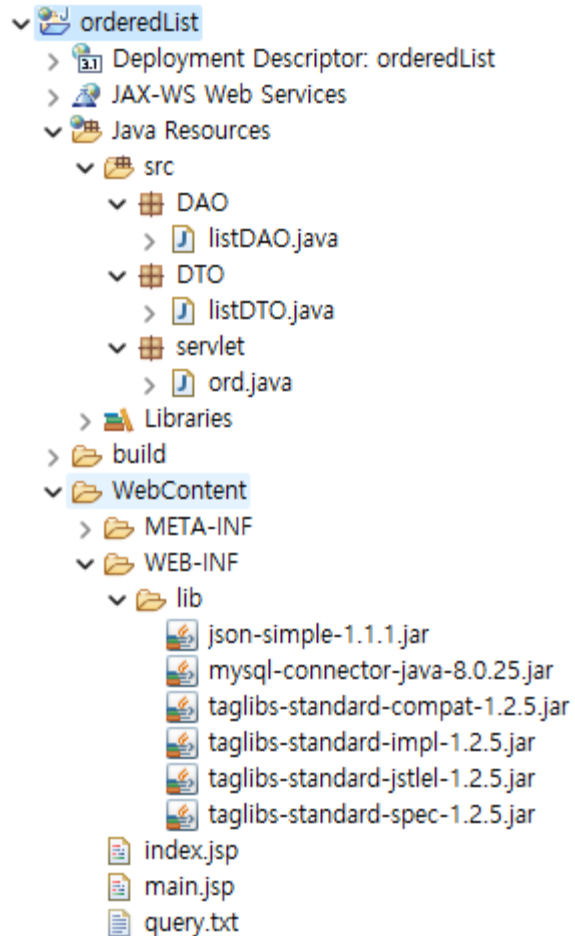
```
1 package servlet;
2
3 import java.io.*;
4 import javax.servlet.*;
5 import javax.servlet.annotation.*;
6 import javax.servlet.http.*;
7
8 @WebServlet("/JQAX")
9 public class JQAX extends HttpServlet {
10   private static final long serialVersionUID = 1L;
11
12   protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
13     System.out.println("통신요청 확인");
14     response.setCharacterEncoding("utf-8");
15     response.getWriter().write("다음값을 반환");
16   }
17
18   protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
19     request.setCharacterEncoding("utf-8");
20     String name=request.getParameter("name");
21     String race=request.getParameter("race");
22     System.out.println("POST 파라미터 전송 확인 : "+name+", "+race);
23     response.setCharacterEncoding("utf-8");
24     response.getWriter().write("포스트방식으로 실행됨을 확인");
25   }
26 }
```

다양한 옵션의 이용이 가능하다.

# 속성값의 종류

**async : Boolean(비동기방식설정)**  
**beforeSend : function(전송전 수행)**  
**complete : function(완료후 수행)**  
**contentType : string(문자 형식 등)**  
**data : JSON, string, array**  
**username : string(접속자 아이디)**  
**password : string(비밀번호)**  
**type : POST, GET(기본값)**  
**success : function(성공시 수행)**  
**timeout : number(시간제한)**

# DRAW AND DROP SORTED LIST



```
create table ordlist(  
    idx int,  
    text text  
);
```

```
insert into ordlist(idx, text) values (1, "JAVA");  
insert into ordlist(idx, text) values (2, "HTML");  
insert into ordlist(idx, text) values (3, "JSP");  
insert into ordlist(idx, text) values (4, "ANDROID");  
insert into ordlist(idx, text) values (5, "JAVASCRIPT");  
insert into ordlist(idx, text) values (6, "PYTHON");  
insert into ordlist(idx, text) values (7, "SQL");
```

## 라이브러리 등을 미리 준비하면 편리

# DRAG AND DROP SORTED LIST

```
<%@ page language="java" contentType="text/html; charset=utf-8"
    pageEncoding="utf-8"%>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>drag and drop</title>
  </head>
  <body>
    <jsp:forward page="ord"></jsp:forward>
  </body>
</html>
```

**index.jsp는 초기 서블릿으로 이동만 진행**

# DRAW AND DROP SORTED LIST

```
<%@ page language="java" contentType="text/html; charset=utf-8"
    pageEncoding="utf-8"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>drag and drop</title>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script src="https://code.jquery.com/ui/1.13.2/jquery-ui.js"></script>
    <style>
  </head>
  <body>
    <div id="wrap">
      <ul class="list-unstyled" id="page_list">
        <c:forEach items="${data}" var="data">
          <li id="${data.idx}" value="${data.text}">${data.text}(${data.idx })</li>
        </c:forEach>
      </ul>
      <input type="hidden" name="page_order_list" id="page_order_list" />
    </div>
    <script>
  </body>
</html>
```

캡데기만 준비해둔다.

# DRAW AND DROP SORTED LIST

```
<style>
  #wrap{
    border:5px solid red;
    width:200px;
    padding:30px;
    margin:0 auto;
  }
  ul{
    margin:0;
    padding:0;
    list-style:none;
  }
  li{
    border:1px solid black;
    margin:5px;
    padding:10px;
    width:150px;
  }
</style>
```

```
<script>
$(document).ready(function(){
  $("#page_list").sortable({
    update : function(event, ui){
      console.log(event); //이동의 원인을 보여줌
      console.log(ui); //누가 이동했는지를 보여줌
      var page_id_array=new Array();
      var page_value_array=new Array();
      $('#page_list li').each(function(){
        page_id_array.push($(this).attr("id"));
        page_value_array.push($(this).attr("value"));
      });
      console.log(page_id_array);
      var str_id=JSON.stringify(page_id_array);
      var str_value=JSON.stringify(page_value_array);
      $.ajax({
        url:"ord",
        method:"POST",
        data:{
          "ids":str_id,
          "values":str_value
        },
        async: false,
        success:function(data){
          console.log(data); //response값이 들어오는 위치임
          console.log("데이터 전송 완료");
        },
        error:function(log){
          console.log("오류발생 : "+log);
        }
      });
    }
  });
});
</script>
```

## JSON타입 전송시 stringify에 주의



# DRAW AND DROP SORTED LIST

```
package servlet;

import java.io.*;
import javax.servlet.*;
import javax.servlet.annotation.*;
import javax.servlet.http.*;
import org.json.simple.*;
import org.json.simple.parser.*;
import java.util.*;
import DTO.listDTO;
import DAO.listDAO;

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

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        System.out.println("doGet호출");

        listDAO DAO=listDAO.getInstance();

        List<listDTO> data=DAO.getList();
        request.setAttribute("data", data);

        RequestDispatcher dispatcher=request.getRequestDispatcher("main.jsp");
        dispatcher.forward(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        response.setCharacterEncoding("utf-8");
        System.out.println("doPost호출");
        String data=request.getParameter("ids");
        System.out.println(data);
        String data2=request.getParameter("values");
        System.out.println(data2);

        JSONParser jParser=new JSONParser();
        try {
            JSONArray obj=(JSONArray)jParser.parse(data);
            JSONArray obj2=(JSONArray)jParser.parse(data2);
            System.out.println(obj.get(0));
            System.out.println(obj2.get(0));
            String[] arr=new String[obj2.size()];
            for(int i=0; i<obj2.size(); i++) {
                arr[i]=(String)obj2.get(i);
            }
            listDAO DAO=listDAO.getInstance();
            DAO.updateList(arr);
        } catch (ParseException e) {
            e.printStackTrace();
        }
        response.getWriter().print("전송확인");//success에서 확인할 반환데이터
    }
}
```

```
package DTO;

public class listDTO {
    private int idx;
    private String text;
    public int getIdx() {
        return idx;
    }
    public void setIdx(int idx) {
        this.idx = idx;
    }
    public String getText() {
        return text;
    }
    public void setText(String text) {
        this.text = text;
    }
}
```

## 서블릿과 오브젝트 준비

# DRAW AND DROP SORTED LIST

```
package DAO;

import java.sql.*;
import java.util.*;

import DTO.listDTO;

public class listDAO {
    private listDAO(){

    }

    private static listDAO instance=new listDAO();|

    public static listDAO getInstance() {
        return instance;
    }
    //커넥션 연결 기능 제작
    public Connection getConnection() throws Exception{
        Connection conn=null;
        String url="jdbc:mysql://127.0.0.1:3306/sample";
        String user="root";
        String pw="iotiot";

        Class.forName("com.mysql.cj.jdbc.Driver");
        conn=DriverManager.getConnection(url, user, pw);

        return conn;
    }
}
```

```
//조회기능
public List<listDTO> getList() {
    List<listDTO> list=new ArrayList<listDTO>();
    String sql="select * from ordlist order by idx asc";
    Connection conn=null;
    PreparedStatement pstmt=null;
    ResultSet rs=null;
    try {
        conn=getConnection();
        pstmt=conn.prepareStatement(sql);
        rs=pstmt.executeQuery();
        while(rs.next()) {
            listDTO DTO=new listDTO();
            DTO.setIdx(rs.getInt("idx"));
            DTO.setText(rs.getString("text"));
            list.add(DTO);
        }
    }catch(Exception e) {
        System.out.println("접속 중 오류 발생 : "+e);
    }finally {
        try {
            if(rs!=null) {rs.close();}
            if(pstmt!=null) {pstmt.close();}
            if(conn!=null) {conn.close();}
        }catch(Exception e) {
            System.out.println("접속 종료 중 오류 발생 : "+e);
        }
    }
    return list;
}
```

```
//수정기능
public void updateList(String[] texts) {
    String sql="update ordlist set idx=? where text=?";
    Connection conn=null;
    PreparedStatement pstmt=null;
    ResultSet rs=null;
    try {
        conn=getConnection();
        for(int i=0; i<texts.length; i++) {
            pstmt=conn.prepareStatement(sql);
            pstmt.setInt(1, i+1);
            pstmt.setString(2, texts[i]);
            System.out.println(pstmt);
            pstmt.executeUpdate();
        }
    }catch(Exception e) {
        System.out.println("접속 중 오류 발생 : "+e);
    }finally {
        try {
            if(rs!=null) {rs.close();}
            if(pstmt!=null) {pstmt.close();}
            if(conn!=null) {conn.close();}
        }catch(Exception e) {
            System.out.println("접속 종료 중 오류 발생 : "+e);
        }
    }
}
```

## 데이터베이스 연결처리