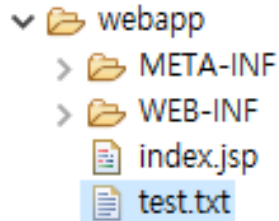


jQuery AJAX
jQuery AJAX

예시

외부 데이터를 내부로 읽어오는 기법



test.txt

1 <h1>텍스트문서에는 다음의 내용이 들어갑니다.</h1>

```
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>동작의 연속처리</title>
8     <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
9   </head>
10  <body>
11    <div id="target"><h2>이 내용이 변경됩니다.</h2></div>
12    <button>외부 자료 읽어오기</button>
13  </body>
14  <script>
15    $("button").click(function(){
16      $("#target").load("test.txt");
17    });
18  </script>
19 </html>
```

XMLHttpRequest의 이용

데이터 선별

외부 데이터를 선택자방식으로 사용 가능

test.txt ✕

```
1 <h1>이 내용은 h1입니다.</h1>
2 <h2>이 내용은 h2입니다.</h2>
3 <h3>이 내용은 h3입니다.</h3>
4 <h4>이 내용은 h4입니다.</h4>
5 <h5>이 내용은 h5입니다.</h5>
6 <h6>이 내용은 h6입니다.</h6>
```

test.txt ✕

```
1 <h1>이 내용은 h1입니다.</h1>
2 <h2>이 내용은 h2입니다.</h2>
3 <h3>이 내용은 h3입니다.</h3>
4 <h4>이 내용은 h4입니다.</h4>
5 <h5>이 내용은 h5입니다.</h5>
6 <h6>이 내용은 h6입니다.</h6>
7
8 <h1>h1이 여러개 존재하는 경우</h1>
9 <h1>다음과 같이 동작합니다.</h1>
```

```
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>동작의 연속처리</title>
8     <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
9   </head>
10  <body>
11    <div id="target"><h2>이 내용이 변경됩니다.</h2></div>
12    <button>외부 자료 읽어오기</button>
13  </body>
14  <script>
15    $("button").click(function(){
16      $("#target").load("test.txt h1");//내부의 특정한 자료를 가져올 수 있다.
17    });
18  </script>
19 </html>
```

이 내용이 변경됩니다.

외부 자료 읽어오기

이 내용은 h1입니다.

h1이 여러개 존재하는 경우
다음과 같이 동작합니다.

외부 자료 읽어오기

받아오는 값을 특정 할 수 있다.

Status의 사용

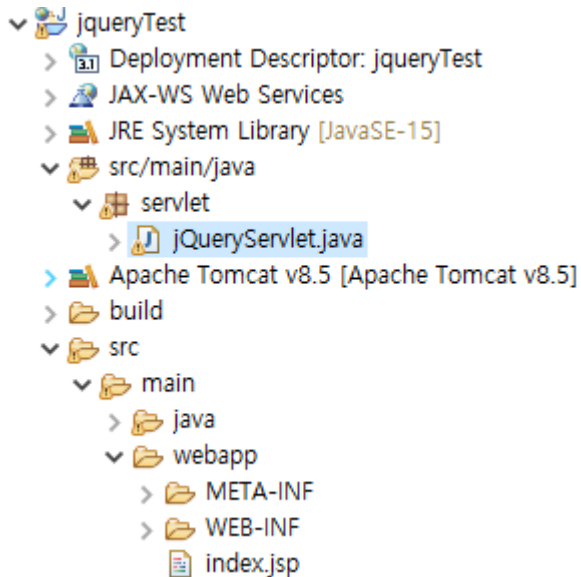
Callback과 status를 확인할 수 있다.

```
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>동작의 연속처리</title>
8     <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
9   </head>
10  <body>
11    <div id="target"><h2>이 내용이 변경됩니다.</h2></div>
12    <button>외부 자료 읽어오기</button>
13  </body>
14  <script>
15    $("button").click(function(){
16      /*$("#target").load("test.txt",function(){
17        alert("동작확인!");
18      }); 종료시의 함수를 호출할 수 있다.*/
19      $("#target").load("test.txt",function(response, status, xhr){
20        console.log("반환값 : "+response); //실패시 오류상태
21        console.log("상태값 : "+status); //실패시 error
22        console.log("XMLHttpRequest : "+xhr); //실패시에도 XMLHttpRequest는 브라우저에 존재하고 있다.
23      });
24    });
25  </script>
26 </html>
```

Javascript의 AJAX방식과
jQuery의 AJAX방식을 비교해 보면 좋다.

AJAX Get 통신

Callback과 status를 확인할 수 있다.



```
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>동작의 연속처리</title>
8     <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
9   </head>
10  <body>
11    <input type="text" placeholder="여기에 값을 입력">
12    <input type="button" value="전송">
13    <h1 id="request"></h1>
14    <h1 id="response"></h1>
15    <h1 id="status"></h1>
16  </body>
17  <script>
18    $("input[type=button]").click(function(){
19      var data=$("input[type=text"]').val();
20      //$.get("jQueryServlet?data="+data);
21      //callback의 사용
22      //서블릿의 response가 작성되어 있어야만 동작
23      $.get("jQueryServlet?data="+data, function(response, status){
24        $("#request").text("전송한 값 : "+data);
25        $("#response").text("전송받은 값 : "+response);
26        $("#status").text("통신상태 값 : "+status);
27      });
28    });
29  </script>
30 </html>
```

Javascript의 AJAX방식과
jQuery의 AJAX방식을 비교해 보면 좋다.

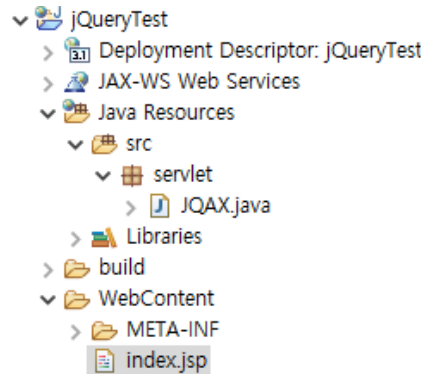
AJAX 서블릿 제작

doGet을 사용한다.

```
1 package servlet;
2
3+ import java.io.IOException;
4
5
6
7
8
9
10 @WebServlet("/jQueryServlet")
11 public class jQueryServlet extends HttpServlet {
12     private static final long serialVersionUID = 1L;
13
14-     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
15         request.setCharacterEncoding("utf-8");
16         String data=request.getParameter("data");
17         System.out.println("취득 데이터 확인"+data);
18
19         response.setCharacterEncoding("utf-8");
20         response.getWriter().write("jQueryServlet이 정상 동작합니다.");
21     }
22-     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
23
24     }
25 }
```

AJAX와 동일한 동작을 보이지만
코드의 길이가 현저히 짧은 것을 확인할 수 있다.

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("다들값을 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 데이터 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)

속성을 적용할 때 문법구조에 주의한다.