

AJAX의 이용

Asynchronous Javascript And XML

AJAX 의 용도?

Asynchronous Javascript And Xml

JSON을 통한 대규모의 데이터의 빠른 전송과
이를 화면의 변화가 없이 읽어오므로
다양한 서비스에서 여러가지 방식으로 사용된다

또한 Javascript라는 언어의 특성 상
다양한 기기에서 동일한 동작으로
사용될 수 있어 편리하다.

**form의 유효성 체크는 물론
동적인 화면의 구현에 매우 편리하다.**

유효성 검증 예시

```
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>검증처리</title>
8   </head>
9   <body>
10    <h1>다음의 내용을 검증합니다.</h1>
11    생일을 입력해 주세요<input type="text" size="10" id="birthDate" onchange="validate()">
12    <div id="dateMessage"></div>
13
14    <script>
51
52 </html>
```

```
<script>
var XHR;
function createXMLHttpRequest(){
  if(window.ActiveXObject){
    XHR=new ActiveXObject("Microsoft.XMLHTTP");
  }else if(window.XMLHttpRequest){
    XHR=new XMLHttpRequest();
  }
}

function validate(){
  createXMLHttpRequest();
  var date=document.getElementById("birthDate");
  var url="ValidationServlet?birthDate="+escape(date.value);
  XHR.open("GET", url, true);
  XHR.onreadystatechange=handleStateChange;
  XHR.send(null);
}

function handleStateChange(){
  if(XHR.readyState==4){
    if(XHR.status==200){
      var mes = XHR.responseXML.getElementsByTagName("message")[0].firstChild.data;
      var val = XHR.responseXML.getElementsByTagName("passed")[0].firstChild.data;
      setMessage(mes, val);
    }
  }
}

function setMessage(message, isValid){
  var messageArea=document.getElementById("dateMessage");
  var fontColor="red";
  if(isValid=="true"){
    fontColor="green";
  }
  messageArea.innerHTML="<span style='color:"+fontColor+"'>" + message + "</span>";
}
</script>
```

화면을 구성한 뒤 서블릿을 제작

유효성 검증 서블릿

```
8 import java.text.*;
9 @WebServlet("/ValidationServlet")
10 public class ValidationServlet extends HttpServlet {
11     private static final long serialVersionUID = 1L;
12     private boolean validateDate(String date) {
13         boolean isValid=true;
14         if(date!=null) {
15             SimpleDateFormat format=new SimpleDateFormat("MM/dd/yyyy");
16             try {
17                 format.parse(date);
18             }catch(Exception e) {
19                 System.out.println("포맷변경 중 오류발생 : "+e);
20                 isValid=false;
21             }
22         }else {
23             isValid=false;
24         }
25         return isValid;
26     }
27     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
28         request.setCharacterEncoding("utf-8");
29         response.setCharacterEncoding("utf-8");
30         response.setContentType("text/xml; charset=utf-8"); //PrintWriter를 가져오기보다 먼저 적어야 한글이 안깨짐
31         PrintWriter out=response.getWriter();
32         boolean passed=validateDate(request.getParameter("birthDate"));
33         String message="잘못된 정보를 입력하셨습니다.";
34         if(passed) {
35             message="올바른 정보를 입력하셨습니다.";
36         }
37         out.println("<response>");
38         out.println("<passed>"+Boolean.toString(passed)+"</passed>");
39         out.println("<message>"+message+"</message>");
40         out.println("</response>");
41         out.close();
42     }
43
44     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
45
46     }
47 }
```

다음의 내용을 검증합니다.

생일을 입력해 주세요

다음의 내용을 검증합니다.

생일을 입력해 주세요

잘못된 정보를 입력하셨습니다.

다음의 내용을 검증합니다.

생일을 입력해 주세요

올바른 정보를 입력하셨습니다.

onkeydown으로 더 편하게 구현가능

HTTP Header?

HyperTextTransportProtocol Header

General Header(일반헤더)

Request/Response Header(요청/응답 헤더)

Entity Header(엔티티 헤더)

세 종류로 구성되어 있다.

**헤더 정보는 body를 포함한 일련의 데이터를
포괄하여 전송하는 역할을 한다.**

HTTP Header 정보 확인

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <readResponseHeader>
3
4 </readResponseHeader>
```

```
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>헤더정보 요청 예시</title>
8   </head>
9   <body>
10    <h1>헤더 정보를 요청합니다.</h1>
11    <a href="javascript:doHeadRequest('allResponseHeader','readResponseHeader.xml')">모든 헤더정보 확인</a><br>
12    <a href="javascript:doHeadRequest('LastModified','readResponseHeader.xml')">마지막 변경시간 확인</a><br>
13    <a href="javascript:doHeadRequest('isResourceAvailable','readResponseHeader.xml')">가용가능한 리소스 확인</a><br>
14    <a href="javascript:doHeadRequest('isResourceAvailable','not-available.xml')">가용불가능한 리소스 확인</a><br>
15    <script>
62   </body>
63 </html>
```

```
<script>
var XHR;
var requestType="";
function createXMLHttpRequest(){
    if(window.ActiveXObject){
        XHR=new ActiveXObject("Microsoft.XMLHTTP");
    }else if(window.XMLHttpRequest){
        XHR=new XMLHttpRequest();
    }
}

function doHeadRequest(request, url){
    requestType=request;
    createXMLHttpRequest();
    XHR.onreadystatechange=handleStateChange;
    XHR.open("HEAD", url, true);
    XHR.send(null);
}

function handleStateChange(){
    if(XHR.readyState==4){
        if(requestType=="allResponseHeader"){
            getAllResponseHeader();
        }else if(requestType=="lastModified"){
            getLastModified();
        }else if(requestType=="isResourceAvailable"){
            getIsResourceAvaliable();
        }
    }
}

function getAllResponseHeader(){
    alert(XHR.getAllResponseHeaders());
    alert(XHR.responseText); //head정보만 얻었으므로 body정보가 존재하지 않는다. XHR.open을 GET방식으로 진행하면 내용이 변경된다.
}

function getLastModified(){
    alert("최종 변경이력 : "+XHR.getResponseHeader("Last-Modified"));
}

function getIsResourceAvaliable(){
    if(XHR.status==200){
        alert("리소스가 가용 가능합니다.");
    }else if(XHR.status==404){
        alert("리소스가 가용 불가능합니다.");
    }else{
        alert("예상치 못한 오류가 발생했습니다. "+XHR.status);
    }
}
}</script>
```

오타에 특히 주의하도록 한다.

HEAD와 GET의 차이

헤더 정보를 요청합니다.

[모든 헤더정보 확인](#)

[마지막 변경시간 확인](#)

[가용가능한 리소스 확인](#)

[가용불가능한 리소스 확인](#)

localhost:8080 내용:

accept-ranges: bytes
content-length: 85
content-type: application/xml
date: Sun, 22 Aug 2021 12:59:38 GMT
etag: W/"85-1629635890307"
last-modified: Sun, 22 Aug 2021 12:38:10 GMT

확인

localhost:8080 내용:

```
<?xml version="1.0" encoding="UTF-8"?>  
<readResponseHeader>  
  
</readResponseHeader>
```

확인

localhost:8080 내용:

최종 변경이력 : Sun, 22 Aug 2021 12:38:10 GMT

확인

localhost:8080 내용:

리소스가 가용 가능합니다.

확인

localhost:8080 내용:

리소스가 가용 불가능합니다.

확인

GET방식을 사용하면 responseText가 존재

화면의 동적 구현

```
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>화면의 비동기 갱신</title>
8   </head>
9   <body>
10    <form action="#">
11      <h3>종류 선택</h3>
12      <select id="race" onchange="refreshList()">
13        <option value="">선택해주세요</option>
14        <option value="dog">강아지</option>
15        <option value="cat">고양이</option>
16        <option value="raccoon">너구리</option>
17      </select>
18      <h3>크기 선택</h3>
19      <select id="size" onchange="refreshList()">
20        <option value="">선택해주세요</option>
21        <option value="small">소형</option>
22        <option value="medium">중형</option>
23        <option value="large">대형</option>
24      </select>
25      <h3>데이터 출력</h3>
26      <select id="list" size="6" style="width:300px;">
27
28      </select>
29    </form>
30    <script>
81 </body>
82 </html>
```

```
<script>
var XHR;
function createXMLHttpRequest(){
  if(window.ActiveXObject){
    XHR=new ActiveXObject("Microsoft.XMLHTTP");
  }else if(window.XMLHttpRequest){
    XHR=new XMLHttpRequest();
  }
}
function createQueryString(race, size){
  var query="race="+race+"&size="+size;
  return query;
}
function clearList(){
  var list=document.getElementById("list");
  while(list.childNodes.length>0){
    list.removeChild(list.childNodes[0]);
  }
}
function updateList(){
  clearList();
  var list=document.getElementById("list");
  var result=XHR.responseXML.getElementsByTagName("pet");
  var option=null;
  for(var i=0; i<result.length; i++){
    option=document.createElement("option");
    option.appendChild(document.createTextNode(result[i].firstChild.nodeValue));
    list.appendChild(option);
  }
}
function handleStateChange(){
  if(XHR.readyState==4){
    if(XHR.status==200){
      updateList();
    }
  }
}
function handleStateChange(){
  if(XHR.readyState==4){
    if(XHR.status==200){
      updateList();
    }
  }
}
function refreshList(){
  var race=document.getElementById("race").value;
  var size=document.getElementById("size").value;
  if(race==" " || size==""){
    clearList();
    return;
  }
  var url="RefreshList?" + createQueryString(race, size);
  createXMLHttpRequest();
  XHR.onreadystatechange=handleStateChange;
  XHR.open("GET", url, true);
  XHR.send(null);
}
</script>
```

서블릿에서 통신을 통해 데이터를 갱신받을 수 있다.

서블릿 제작

```
1 package servlet;
2
3 public class PetList {
4     private String race;
5     private String size;
6     private String name;
7
8     public PetList(String r, String s, String n) {
9         race=r;
10        size=s;
11        name=n;
12    }
13 }
```

```
3 import java.io.*;
4 import javax.servlet.*;
5 import javax.servlet.annotation.*;
6 import javax.servlet.http.*;
7 import java.util.*;
8
9 @WebServlet("/RefreshList")
10 public class RefreshList extends HttpServlet {
11     private static List<Pet> list=new ArrayList<Pet>();
12     private static final long serialVersionUID = 1L;
13     public void init() throws ServletException{
14         list.add(new Pet("dog", "small", "말티즈"));
15         list.add(new Pet("dog", "small", "시추"));
16         list.add(new Pet("dog", "medium", "그레이하운드"));
17         list.add(new Pet("dog", "medium", "보르조이"));
18         list.add(new Pet("dog", "large", "골든리트리버"));
19         list.add(new Pet("dog", "large", "보더콜리"));
20         list.add(new Pet("cat", "small", "싱가푸라"));
21         list.add(new Pet("cat", "small", "먼치킨"));
22         list.add(new Pet("cat", "medium", "터키시왕고리"));
23         list.add(new Pet("cat", "medium", "통키니즈"));
24         list.add(new Pet("cat", "large", "메인쿤"));
25         list.add(new Pet("cat", "large", "벵갈"));
26         list.add(new Pet("raccoon", "small", "작은너구리"));
27         list.add(new Pet("raccoon", "medium", "너구리"));
28         list.add(new Pet("raccoon", "large", "큰너구리"));
29     }
30     protected void processRequest(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException{
31         response.setContentType("text/html; charset=utf-8");
32         String race=request.getParameter("race");
33         String size=request.getParameter("size");
34         StringBuffer result=new StringBuffer("<pets>");
35         for(Pet p:list) {
36             if(p.getRace().equals(race)) {
37                 if(p.getSize().equals(size)) {
38                     result.append("<pet>");
39                     result.append(p.getName());
40                     result.append("</pet>");
41                 }
42             }
43         }
44         result.append("</pets>");
45         System.out.println(result);
46         response.setContentType("text/xml");
47         response.getWriter().write(result.toString());
48     }
49     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
50         processRequest(request, response);
51     }
52 }
53 }
```

패러미터에 따라 다른 결과를 전송

결과 확인

종류 선택

선택해주세요 ▼

크기 선택

선택해주세요 ▼

데이터 출력

종류 선택

강아지 ▼

크기 선택

중형 ▼

데이터 출력

그레이하운드
보르조이

종류 선택

고양이 ▼

크기 선택

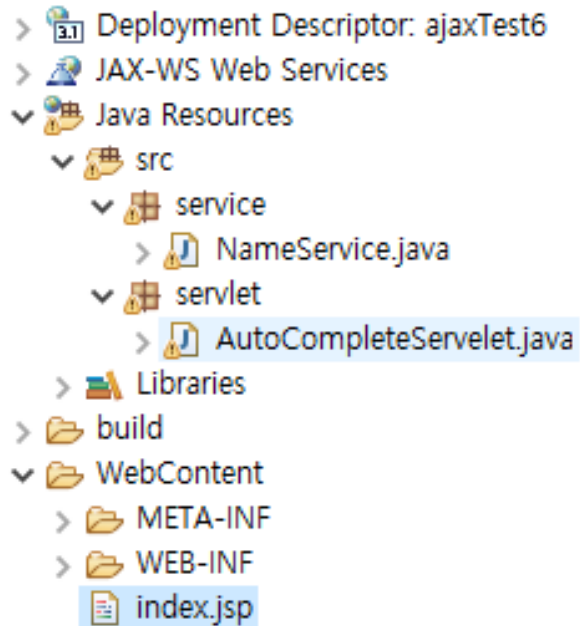
대형 ▼

데이터 출력

메인쿤
벵갈

화면이 변경되는 데이터를 받아오고 있음

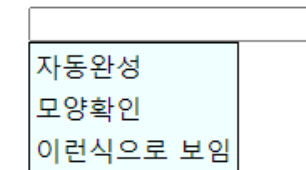
검색어 자동완성



```
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>자동완성(추천검색어) 제작</title>
8     <style>
18   </head>
19   <body>
20     <h1>자동완성 기능 구현</h1>
21     <input type="text" size="20" id="names" onkeyup="findNames()">
22     <div id="popup">
23       <table id="nameTable">
24         <tbody id="nameTableBody">
25           <tr>
26             <td>자동완성</td>
27           </tr>
28           <tr>
29             <td>모양확인</td>
30           </tr>
31           <tr>
32             <td>이런식으로 보임</td>
33           </tr>
34         </tbody>
35       </table>
36     </div>
37     <script>
110   </body>
111 </html>
```

```
<style>
  .mouse:hover{
    background-color:black;
    color:white;
  }
  #nameTable{
    background-color:azure;
    border:1px solid black;
  }
</style>
```

자동완성 기능 구현



검색어의 추천 또는 이전의 검색내역을 알릴 수 있음

스크립트 작성

```
<script>
var XHR;
var div;
var input;
var nameTable;
var nameTableBody;
function createXMLHttpRequest(){
    if(window.ActiveXObject){
        XHR=new ActiveXObject();
    }else if(window.XMLHttpRequest){
        XHR=new XMLHttpRequest();
    }
}
function init(){
    input=document.getElementById("names");
    nameTable=document.getElementById("nameTable");
    nameTableBody=document.getElementById("nameTableBody");
    div=document.getElementById("popup");
}
function findNames(){
    init();
    if(input.value.length>0){
        createXMLHttpRequest();
        var url="AutoCompleteServlet?names="+escape(input.value);
        XHR.open("GET", url, true);
        XHR.onreadystatechange=handleStateChange;
        XHR.send(null);
    }else{
        clearNames();
    }
}
```

```
function clearNames(){
    var ntb=nameTableBody.childNodes.length;
    for(var i=ntb-1; i>=0; i--){
        nameTableBody.removeChild(nameTableBody.childNodes[i]);
    }
    div.style.border="none";
}
function populateName(cell){
    input.value=cell.firstChild.nodeValue;
    clearNames();
}
function setNames(the_names){
    clearNames();
    var size=the_names.length;
    var row;
    var cell;
    var txtNode;
    for(var i=0; i<size; i++){
        var nextNode=the_names[i].firstChild.data;
        row=document.createElement("tr");
        cell=document.createElement("td");
        cell.setAttribute("class", "mouse");
        cell.setAttribute("bgcolor", "azure");
        cell.onclick=function(){populateName(this);};
        txtNode=document.createTextNode(nextNode);
        cell.appendChild(txtNode);
        row.appendChild(cell);
        nameTableBody.appendChild(row);
    }
}
function handleStateChange(){
    if(XHR.readyState==4){
        if(XHR.status==200){
            setNames(XHR.responseXML.getElementsByTagName("name"));
        }else if(XHR.status==204){
            //리소스는 존재하지만 데이터가 없는 경우, 서버릿에서 지정할 수 있음
            clearNames();
        }
    }
}
```

기존과 같은 Get방식의 전송을 진행하고 있음

서비스 작성

```
1 package service;
2
3 import java.util.*;
4
5 public class NameService {
6     private List names;
7     private NameService(List list_of_names) {
8         this.names=list_of_names;
9     }
10    public static NameService getInstance(List list_of_names) {
11        return new NameService(list_of_names);
12    }
13    public List findNames(String prefix) {
14        System.out.println("입력확인 : "+prefix);
15        String prefix_upper=prefix.toUpperCase();
16        List matches=new ArrayList();
17        Iterator itr=names.iterator();
18        while(itr.hasNext()) {
19            String name=(String)itr.next();
20            String name_upper_case=name.toUpperCase();
21            if(name_upper_case.startsWith(prefix_upper)) {
22                boolean result=matches.add(name);
23            }
24        }
25        return matches;
26    }
27 }
```

서블릿의 서비스를 진행해줄 class를 작성

서블릿 작성

```
1 package servlet;
2
3 import java.io.*;
4 import javax.servlet.*;
5 import javax.servlet.annotation.*;
6 import javax.servlet.http.*;
7 import java.util.*;
8 import service.NameService;
9
10 @WebServlet("/AutoCompleteServlet")
11 public class AutoCompleteServlet extends HttpServlet {
12     private List names=new ArrayList();
13     public void init(ServletConfig config)throws ServletException {
14         names.add("Abe");
15         names.add("Abel");
16         names.add("Abigail");
17         names.add("Abner");
18         names.add("Abraham");
19         names.add("Marcus");
20         names.add("Marcy");
21         names.add("Marge");
22         names.add("Marie");
23     }
24     private static final long serialVersionUID = 1L;
25
26     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
27         String prefix=request.getParameter("names");
28         NameService service=NameService.getInstance(names);
29         List matching=service.findNames(prefix);
30         if(matching.size()>0) {
31             PrintWriter out=response.getWriter();
32             response.setContentType("text/xml");
33             response.setHeader("Cache-control", "no-cache");
34             out.println("<response>");
35             Iterator itr=matching.iterator();
36             while(itr.hasNext()) {
37                 String name=(String)itr.next();
38                 out.println("<name>"+name+"</name>");
39             }
40             out.println("</response>");
41             matching=null;
42             service=null;
43             out.close();
44         }else {
45             response.setStatus(HttpServletResponse.SC_NO_CONTENT);
46         }
47     }
48
49     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
50         doGet(request, response);
51     }
52     public String getServletInfo() {
53         return "서블릿 정보";
54     }
55
56 }
```

자동완성 확인

자동완성 기능 구현

자동완성
모양확인
이런식으로 보임

자동완성 기능 구현

Abe
Abel
Abigail
Abner
Abraham

자동완성 기능 구현

자동완성 기능 구현

Marcus
Marcy
Marge
Marie

한글은 초/중/종성을 나누어 검색해야 한다.