

Ajax Support Service

Summary

In general, the Ajax function is developed in Javascript language but may be a task not easy for J2EE developers who are familiar with server-side implementation.

The Ajax support service provides the frequently used function in the form of custom tag using Ajax. The function uses the open source library, AjaxTags.

Description

Installation

System Environment and Required Library

- JDK 1.5
- Servlet container running Servlets 2.4+ and JSP 2.0+ (jsp-api 2.0, servlet-api 2.4)
- [AjaxTags Library](#)

Installation Order

1. Go to [AjaxTags Download Site](#) to download the relevant library and locate in WEB-INF/lib.
2. Set web.xml.
3. `<servlet>`
4. `<servlet-name>sourceloader</servlet-name>`
5. `<servlet-class>net.sourceforge.ajaxtags.servlets.SourceLoader</servlet-class>`
- 6.
7. `<init-param>`
8. `<param-name>prefix</param-name>`
9. `<param-value>/ajaxtags</param-value>`
10. `</init-param>`
11. `</servlet>`
- 12.
13. `<servlet-mapping>`
14. `<servlet-name>sourceloader</servlet-name>`
15. `<url-pattern>/ajaxtags/js/*</url-pattern>`
16. `</servlet-mapping>`
- 17.
18. `<servlet-mapping>`
19. `<servlet-name>sourceloader</servlet-name>`
20. `<url-pattern>/ajaxtags/img/*</url-pattern>`
21. `</servlet-mapping>`
- 22.
23. `<servlet-mapping>`
24. `<servlet-name>sourceloader</servlet-name>`
25. `<url-pattern>/ajaxtags/css/*</url-pattern>`
- `</servlet-mapping>`

AjaxTags Tag Reference

ajax:autocomplete

This is the AutoCompletefunction. If you enter prefix characters in a search box, you can see relevant recommended search terms.

Parameter	Description	Required or not
baseUrl	URL for server-side action that shows the result data for auto-complete function	yes
source	Name of text file to show the recommended list of search terms.To	yes

	show the recommended list of search in the input field, enter target and source same.	
target	Text field name entered by the user.	yes
parameters	Parameters to add to baseUrl. If there are several parameters, separate with comma.	yes
className	CSS class name to apply to recommended search list	yes
indicator	Indicator to display during Ajax request.	no
minimumCharacters	Minimum input value for Ajax request.	no
preFunction	Name of function that operates before Ajax request starts.	no
postFunction	Name of function that operates after Ajax request is completed.	no
errorFunction	Function name that operates during Ajax request error.	no

ajax:select

Change the value on select box and configure the list with the related values in other select box. Linked SelectBox.

Parameter	Description	Required or not
baseUrl	URL for server-side action that sends result data for the autocomplete function	yes
source	Name of text field to show the recommended list of search words. To show recommended search list in the input field, enter target and source same.	yes
target	Text field name entered by the user.	yes
parameters	Parameters to add to baseUrl. If there are several parameters, separate with comma.	no
eventType		no
executeOnLoad	Whether to separately display the execution when executing the select box with response data.[default=false]	no
defaultOptions	Basic list to shown when there is no Ajax response value. Write out by separating with comma.	no
preFunction	Name of function that operates before Ajax request starts.	no
postFunction	Name of function that operates after Ajax request is completed.	no
errorFunction	Function name that operates during Ajax request error.	no
parser	Parser for response data.[default=ResponseHtmlParser]	no

ajax:tabPanel

To show pages that consists of tab without refresh.

Parameter	Description	Required or not
id	ID of tabPanel	yes
preFunction	Name of function that operates before Ajax request starts.	no
postFunction	Name of function that operates after Ajax request is completed.	no
errorFunction	Function name that operates during Ajax request error.	no
parser	parser for response data.[default=ResponseHtmlParser]	no

others

Besides this, there are several functions. Check the following [AjaxTags Site](#) for Tag reference of AjaxTags and how to use it.

Common development task

Following tasks occur in common no matter which tag of AjaxTags is used.

JSP

Declare Tag Library

```
<%@ taglib prefix="ajax" ri="http://ajaxtags.sourceforge.net/tags/ajaxtags" %>
```

Declare Javascript, CSS

```
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/prototype.js"></script>
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/scriptaculous/scriptaculous.js"></script>
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/overlibmws/overlibmws.js"></script>
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/ajaxtags.js"></script>
<link type="text/css" rel="stylesheet"
href="<%=request.getContextPath()%>/ajaxtags/css/ajaxtags.css" />
<link type="text/css" rel="stylesheet"
href="<%=request.getContextPath()%>/ajaxtags/css/displaytag.css" />
```

Controller

To use AjaxTags, the result data should have data format (XML style) in AjaxTags. For this purpose, AjaxTags provides API for processing of data called AjaxXmlBuilder. While the task of converting the result data using AjaxXmlBuilder can be done at View, the number of view can be added by the unit of function. In this regard, it is recommended to put in the Model object after converting at Controller and send it to View, and to use one View in common.

org.ajaxtags.helpers.AjaxXmlBuilder

Add the Result of Collection Type at Once

```
List<Department>deptList = departmentService.getDepartmentList(param);
String result = new AjaxXmlBuilder().addItem(deptList, "deptname", "deptid").toString();
model.addObject("ajaxXml", ajaxXmlBuilder.toString());
```

Add Case One by One

```
List<Department>deptList = departmentService.getDepartmentList(param);
AjaxXmlBuilder ajaxXmlBuilder = new AjaxXmlBuilder();
for (Iterator iter = deptList.iterator(); iter.hasNext();) {
    Department dept = (Department) iter.next();
    ajaxXmlBuilder.addItem(dept.getDeptname(), dept.getDeptid());
}
model.addObject("ajaxXml", ajaxXmlBuilder.toString());
```

Result data is same.

```
<?xml version="1.0" encoding="UTF-8"?>
<ajax-response>
<response>
<item>
<name>Lunch Menu Planning Team</name>
<value>1200</value>
</item>
<item>
```

```

<name>Night Overtime Prohibition Capacity Team</name>
<value>1300</value>
</item>
...
</response>
</ajax-response>

```

View

Since it is not general response type printed on JSP page, common view should be created for response processing.

Set the form of result data(XML) in response object.

Take out and write the result data of Model object sent by controller.

```
packagecom.easycompany.view;
```

```

importjava.io.PrintWriter;
importjava.util.Map;
importjavax.servlet.http.HttpServletRequest;
importjavax.servlet.http.HttpServletResponse;
importorg.springframework.web.servlet.view.AbstractView;

```

```
public class AjaxXmlView extends AbstractView {
```

```
    @Override
```

```
    protected void renderMergedOutputModel(Map model,
        HttpServletRequest request, HttpServletResponse response)
        throwsException {
```

```

        response.setContentType("text/xml");
        response.setHeader("Cache-Control", "no-cache");
        response.setCharacterEncoding("UTF-8");

```

```

        PrintWriter writer = response.getWriter();
        writer.write((String) model.get("ajaxXml")); //Model Attribute name is used in

```

```
common...
```

```
        writer.close();
```

```
    }
```

```
}
```

Example

ajax:autocomplete

At the employee information inquiry page, let's apply the autocomplete function in the name field, one of inquiry conditions.

To start entering the name to search, the name with the prefix corresponding to input value appears as recommended list.

사원번호 :	<input type="text"/>	부서번호 :	<input type="text"/>	이름 :	<input type="text" value="김"/> <ul style="list-style-type: none"> 김길동 김나리 김길수 김앤젤 	<input type="button" value="검색"/>
--------	----------------------	--------	----------------------	------	---	-----------------------------------

	사원번호	부서번호	이름	나이	이메일
1	1200	1200	김길동	28	kkd@e
2	1100	1100	김길수	39	kks@easycompany.com
3	1200	1200	강감찬	17	kkc@easycompany.com

JSP

/easycompany/webapp/WEB-INF/jsp/employeeelist.jsp

```

...
<%@ taglib prefix="ajax" uri="http://ajaxtags.sourceforge.net/tags/ajaxtags" %>
...
<!--Ajax Tags-->
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/prototype.js"></script>
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/scriptaculous/scriptaculous.js"></script>
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/overlibmws/overlibmws.js"></script>
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/ajaxtags.js"></script>
<link type="text/css" rel="stylesheet"
href="<%=request.getContextPath()%>/ajaxtags/css/ajaxtags.css" />
<link type="text/css" rel="stylesheet"
href="<%=request.getContextPath()%>/ajaxtags/css/displaytag.css" />
...
<form:formcommandName="searchCriteria" action="/easycompany/employeeList.do">
<table width="50%" border="1">
  <tr>
    <td>
      <td>Employee Number : <form:input path="searchEid"/></td>
      <td>Department Number : <form:input path="searchDid"/></td>
      <td>Name : <form:input path="searchName"/>
      </td>
      <td><input type="submit" value="Search"
onclick="this.disabled=true,this.form.submit();" /></td>
    </tr>
  </table>
</form:form>

<ajax:autocomplete
baseUrl="{pageContext.request.contextPath}/suggestName.do"
source="searchName"
target="searchName"
className="autocomplete"
minimumCharacters="1" />
...

```

Controller

```

com.easycompany.controller.annotation.AjaxController

packagecom.easycompany.controller.annotation;
...
importnet.sourceforge.ajaxtags.xml.AjaxXmlBuilder;
importcom.easycompany.view.AjaxXmlView;

@Controller
public class AjaxController {

    @Autowired
    privateEmployeeServiceemployeeService;

    @Autowired
    privateDepartmentServicedepartmentService;

    @RequestMapping("/suggestName.do")
    protectedModelAndViewsuggestName(@RequestParam("searchName") StringsearchName){

        ModelAndView model = new ModelAndView(new AjaxXmlView());
        List<String>nameList = employeeService.getNameListForSuggest(searchName);

        AjaxXmlBuilderajaxXmlBuilder = new AjaxXmlBuilder();

        for(Stringname:nameList){
            ajaxXmlBuilder.addItem(name, name, false);
        }
        model.addObject("ajaxXml",ajaxXmlBuilder.toString());
        return model;
    }
}

```

Korean Processing Setting

From above example, employee name prefix value is transferred to the parameter as shown in '/suggestName.do?searchName=Kim'. To process properly when the parameter value is Korean, encoding processing is required in {Tomcat DIR}/conf/server.xml. For UTF-8 encoding, URIEncoding="utf-8" needs to be added to <Connector/>tag.

```

...
<Connector port="8080" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" redirectPort="8443" acceptCount="100"
connectionTimeout="20000" disableUploadTimeout="true" URIEncoding="utf-8"/>
...
<Connector port="8009"
enableLookups="false" redirectPort="8443" protocol="AJP/1.3" URIEncoding="utf-8" />

```

ajax:select

In employee information modification(input) page, select one department at the upper department information select box, and the lower department information select box forms the option with lower department information that belongs to relevant upper department.

사원번호	6
부서번호	<div style="border: 1px solid red; padding: 2px;"> 경영기획실 마근금지역량팀 </div>
이름	<input type="text" value="재기찬"/>
비밀번호	<input type="text" value="6"/>
주민번호	<input type="text"/> - <input type="text"/>
나이	<input type="text" value="34"/>
이메일	<input type="text" value="jackie@easycompany.com"/>

JSP

/easycompany/webapp/WEB-INF/jsp/addemployee.jsp, /easycompany/webapp/WEB-INF/jsp/modifyemployee.jsp

```

...
<%@ taglib prefix="ajax" uri="http://ajaxtags.sourceforge.net/tags/ajaxtags" %>
...
<!--Ajax Tags-->
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/prototype.js"></script>
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/scriptaculous/scriptaculous.js"></script>
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/overlibmws/overlibmws.js"></script>
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/ajaxtags.js"></script>
<link type="text/css" rel="stylesheet"
href="<%=request.getContextPath()%>/ajaxtags/css/ajaxtags.css" />
<link type="text/css" rel="stylesheet"
href="<%=request.getContextPath()%>/ajaxtags/css/displaytag.css" />
...
<form:formcommandName="employee">
...
    <tr>
        <th>Department number</th>
        <td>
            <form:select path="superdeptid">
                <option value="">Select the upper department.</option>
                <form:options items="{deptInfoOneDepthCategory}" />
            </form:select>
            <form:select path="departmentid">
                <option value="">Select the service department.</option>
                <form:options items="{deptInfoTwoDepthCategory}" />
            </form:select>
        </td>
    </tr>
...
</form:form>

<ajax:select
baseUrl="{pageContext.request.contextPath}/autoSelectDept.do"

```

```

parameters="depth=2,superdeptid={superdeptid}"
source="superdeptid"
target="departmentid"
emptyOptionName="Select model"/>
...

```

ajax:tabPanel, ajax:tab

When showing the lower department list that belongs to upper department at department information page, it shows them in tab.

[경영기획실](#) | [경영지원실](#) | [IT연구소](#) | [공공사업부](#) | [금융사업부](#)

부서번호	부서명	상위부서명
1100	회식메뉴혁신팀	경영기획실
1200	점심메뉴기획팀	경영기획실
1300	야근금지역량팀	경영기획실
1400	사랑의짜대기팀	경영기획실

JSP

/easycompany/webapp/WEB-INF/jsp/departmentlist.jsp

```

...
<%@ taglib prefix="ajax" uri="http://ajaxtags.sourceforge.net/tags/ajaxtags" %>
...
<!--Ajax Tags-->
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/prototype.js"></script>
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/scriptaculous/scriptaculous.js"></script>
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/overlibmws/overlibmws.js"></script>
<script type="text/javascript"
src="<%=request.getContextPath()%>/ajaxtags/js/ajaxtags.js"></script>
<link type="text/css" rel="stylesheet"
href="<%=request.getContextPath()%>/ajaxtags/css/ajaxtags.css" />
<link type="text/css" rel="stylesheet"
href="<%=request.getContextPath()%>/ajaxtags/css/displaytag.css" />
...
    <ajax:tabPanel id="departmentTab">
        <c:forEach items="{departmentlist}" var="departmentinfo" varStatus="status">
            <c:choose>
                <c:when test="{status.first}">
                    <ajax:tab caption="{departmentinfo.deptname}"
baseUrl="/easycompany/subDepartmentList.do?superdeptid={departmentinfo.deptid}&depth=2"
                    defaultTab="true"/>
                </c:when>
                <c:otherwise>
                    <ajax:tab caption="{departmentinfo.deptname}"
baseUrl="/easycompany/subDepartmentList.do?superdeptid={departmentinfo.deptid}&depth=2"/>
                </c:otherwise>
            </c:choose>
        </c:forEach>
    </ajax:tabPanel>

```



```
        </c:choose>  
</c:forEach>  
</ajax:tabPanel>
```

...

Reference

[AjaxTags Web Site](#)