

Applies To:

SAP Exchange Infrastructure XI 3.0 (SP11 and higher)

Summary

This code sample provides a simple yet useful HTML interface for sending XML data to XI 3.0 via HTTP.

By: James Guanzon

Company: SAP Labs, LLC

Date: 28 March 2006

Table of Contents

Applies To:	1
Summary	1
Table of Contents	1
Use Case.....	2
Specific Applications	2
Sample Code for the HTTP Client for XI 3.0	4
Keywords.....	8
Author Bio.....	9
Disclaimer & Liability Notice.....	9

Use Case

When you require an **input / output** mechanism for communication with **SAP Exchange Infrastructure 3.0 (XI)**, the following HTML code can be used as a simple application with a user interface, providing user-defined input for receiving output from XI in a normal HTTP transaction. Yes! Finally, you have a simple and easy-to-use application from SAP. Best of all it's free...

Specific Applications

The code that follows is specifically designed for use with the SAP Developer Network ([SDN](#)) weblog entitled "[How to Support Industry Standards in XI](#)" and the communication to the SAP Exchange Infrastructure 3.0.

Please cut and paste the HTML code sample to a file on your local file system. In this example, the HTML code will be saved to a file named "HTTPClient__OAG_ProcessPurchaseOrder.html".

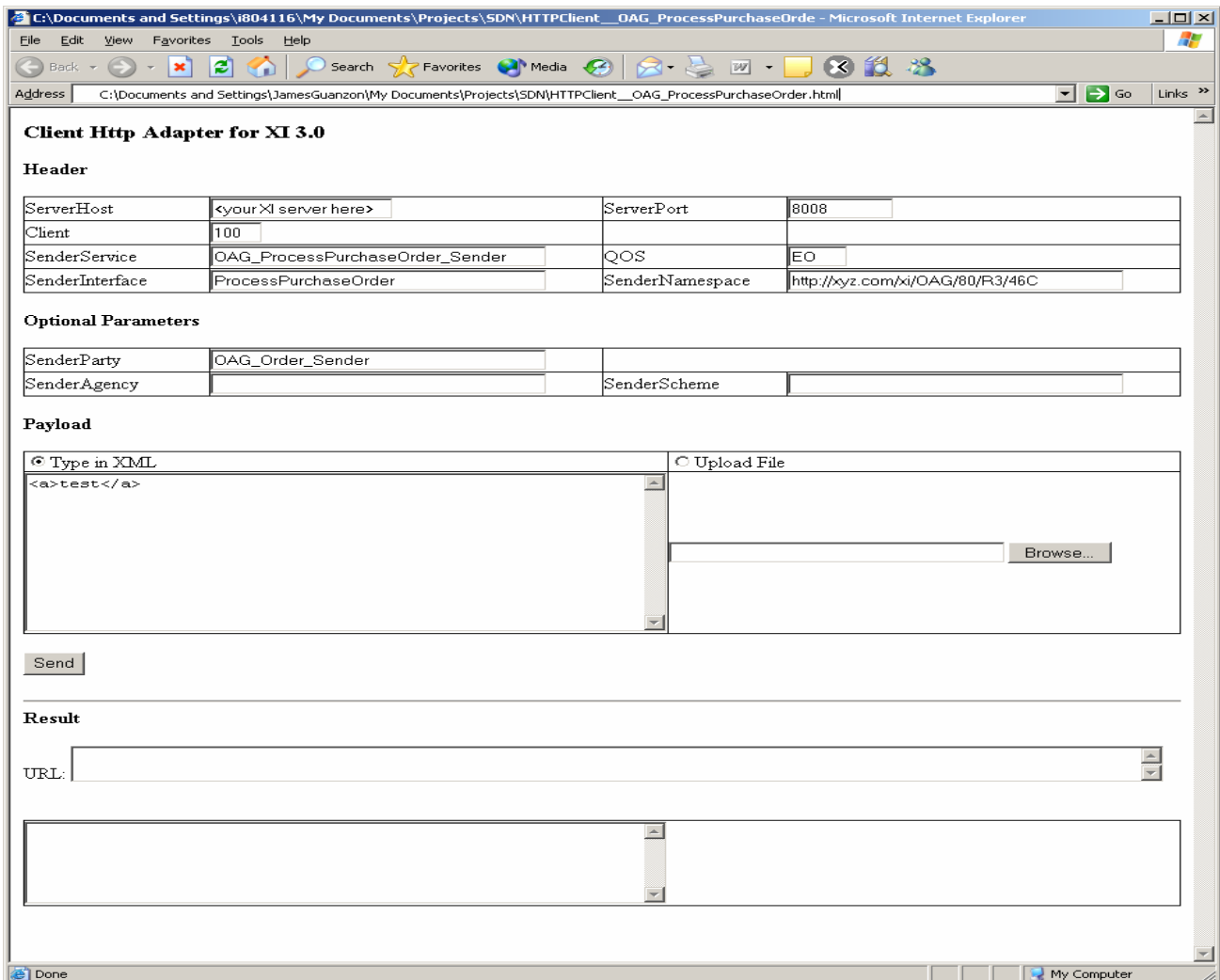
A number of parameters must be entered in order to communicate via HTTP to XI 3.0.

Parameter	Value	Notes
ServerHost	<your XI server here>	Your XI Server Host. Example: nspad375.pal.sap.corp
Client	<your client number here>	Typically 3 digits. Contact your SAP administrator for details. Example: 100
ServerPort	<your XI server port here>	Contact your SAP administrator for details. Example: 8008
SenderService	OAG_ProcessPurchaseOrder_Sender	Your XI Sender Service value
SenderInterface	ProcessPurchaseOrder	Your XI Sender Interface value
QOS	EO	Possible values are: EO – Exactly Once BE – Best Effort EOIO – Exactly Once In Order
SenderNamespace	http://xyz.com/xi/OAG/80/R3/46C	The configured XI Sender Namespace.
SenderParty	OAG_Order_Sender	The configured XI Sender party.

HTTP Client Sample Code for Communicating with XI 3.0



Parameter	Value	Notes
SenderAgency		Optional parameter. Can remain blank.
SenderScheme		Optional parameter. Can remain blank.
Radio Button: Type in XML	Deselect.	Default selection. Can enter any text as input.
Radio Button: Upload File	Select.	Upload the file created in this SDN weblog: XI – A Heavyweight Alternative to Notepad



Sample Code for the HTTP Client for XI 3.0

This is an example of a code sample:

```
<html>
<!-- HTTP Client for XI 3.0 -->
<!-- james.guanzon@sap.com sap labs, llc 03.28.2006 -->

<script type="text/javascript">
<!--
function button1_onclick() {
    var result = "Result:  ";
    var payload = "<?xml version=\"1.0\" encoding=\"UTF-8\" ?>";
    // escape "http://"
    var senderNamespace =
escape(document.MessageParameters.SenderNamespace.value);

    var reqString = "http://"
    reqString = reqString + document.MessageParameters.Server.value+":";
    reqString = reqString + document.MessageParameters.Port.value +
"/sap/xi/adapter_plain?";
    reqString = reqString + "namespace=" + senderNamespace;
    reqString = reqString + "&interface=" +
document.MessageParameters.SenderInterface.value;
    reqString = reqString + "&service=" +
document.MessageParameters.SenderService.value;
    reqString = reqString + "&party=" +
document.MessageParameters.SenderParty.value;
    reqString = reqString + "&agency=" +
document.MessageParameters.SenderAgency.value;
    reqString = reqString + "&scheme=" +
document.MessageParameters.SenderScheme.value;
    reqString = reqString + "&QOS=" + document.MessageParameters.Qos.value;
    reqString = reqString + "&sap-user=xiappluser&sap-password=xipass";
    reqString = reqString + "&sap-client=" +
document.MessageParameters.Client.value + "&sap-language=EN";

    var xhttp = new XMLHttpRequest("msxml2.xmlhttp");
    xhttp.open ("POST", reqString, false);
```

```
document.MessageParameters.URL.value=reqString;

if (document.MessageParameters.Source[0].checked == true) {
    payload = document.MessageParameters.xmlData.value;
    xhttp.send (payload);
}
else{
    var xmlDoc = new ActiveXObject("microsoft.xmlDOM");
    xmlDoc.async=false;
    xmlDoc.load (document.MessageParameters.xmlFile.value);
    xhttp.send (xmlDoc);
}
result = result + "\n" + xhttp.responseText;
xhttp.close;
document.MessageParameters.response.value=result;
}
//-->
</script>
<head></head>

<body>

<h3>Client Http Adapter for XI 3.0</h3>
<form name="MessageParameters">
    <table border="1" cellpadding="0" cellspacing="0" style="border-collapse:
collapse" bordercolor="#111111" width="100%">
        <h4>Header</h4>
        <tbody>
            <tr>
                <td width="10%"><label>ServerHost</label> </td>
                <td width="22%"><input type="text" id="host" name="Server"
value="<your XI server here>" size="20" /> </td>
                <td width="10%"><label>ServerPort</label> </td>
                <td width="22%"><input type="text" id="port" name="Port" value="8008"
size="10" /> </td>
```

```
</tr>
<tr>
  <td width="10%">Client</td>
  <td width="22%"><input type="text" id="client" name="Client"
value="100" size="3" /></td>
  <td width="10%">&nbsp;</td>
  <td width="22%">&nbsp;</td>
</tr>
<tr>
  <td width="10%"><label>SenderService</label> </td>
  <td width="22%">
    <input type="text" id="senderService"
name="SenderService" value="OAG_ProcessPurchaseOrder_Sender" size="40"
/> </td>
  <td width="10%">QOS</td>
  <td width="22%">
    <input type="text" id="qos" name="Qos" value="EO" size="4" /> </td>
</tr>
<tr>
  <td width="10%"><label>SenderInterface</label> </td>
  <td width="22%">
    <input type="text" id="senderInterface"
name="SenderInterface" value="ProcessPurchaseOrder" size="40" /> </td>
  <td width="10%"><label>SenderNamespace</label> </td>
  <td width="22%">
    <input type="text" id="senderNamespace"
name="SenderNamespace" value="http://xyz.com/xi/OAG/80/R3/46C"
size="40" /></td>
</tr>
</tbody>
</table>
&#160;
  <table border="1" cellpadding="0" cellspacing="0" style="border-collapse:
collapse" bordercolor="#111111" width="100%">
    <h4>Optional Parameters</h4>
```

```
<tbody>
<tr>
  <td width="10%">SenderParty</td>
  <td width="22%"><input type="text" id="senderParty" name="SenderParty"
value="OAG_Order_Sender"
  size="40" /> </td>
</tr>
<tr>
  <td width="10%">SenderAgency</td>
  <td width="22%"><input type="text" id="senderAgency"
name="SenderAgency"
  size="40" /> </td>
  <td width="10%">SenderScheme</td>
  <td width="22%"><input type="text" id="senderScheme"
name="SenderScheme"
  size="40" /> </td>
</tr>
</tbody>
</table>
&#160;
<table border="1" cellpadding="0" cellspacing="0" style="border-collapse:
collapse" bordercolor="#111111" width="100%">
<h4>Payload</h4>
<tbody>
<tr>
  <fieldset style="padding: 2">
    <td width="50%"><input type="radio" name="Source" value="Textarea"
checked="checked" />Type in XML</td>
    <td width="50%"><input type="radio" name="Source" value="File"
/>Upload File</td>
  </fieldset>
</tr>
<tr>
  <td width="50%"><textarea name="xmlData" rows="10"
cols="60">&lt;a&gt;test&lt;/a&gt;</textarea></td>
  <td width="50%"><input type="file" name="xmlFile" size="40" /> </td>
</tr>
</tbody>
</table>
```

```
</tr>
</tbody>
</table>
<p>
  <input type="button" value="Send" id="button1" name="button1"
LANGUAGE="javascript" onclick="button1_onclick()" />
</p>
<hr />
<table border="1" cellpadding="0" cellspacing="0" style="border-collapse:
collapse" bordercolor="#111111" width="100%">
  <h4 align="left">Result</h4>
  <p align="left">URL: <textarea rows="2" name="URL"
cols="104"></textarea></p>
  <p align="left">&nbsp;</p>
  <tbody>
    <tr>
      <td width="50%"><textarea name="response" rows="5"
cols="60"></textarea></td>
    </tr>
  </tbody>
</table>
</form>

</body>

</html>
```

Keywords

How to communicate with XI

HTTP Client

HTML Client

Alternative to Quick Test Tool in XI

Author Bio



James Guanzon is a developer at SAP Labs, LLC in Palo Alto, California. James has worked for SAP Labs, LLC since 2001. He works in the SAP Industry Solutions Development organization for the **IndustrySpeak** Program, which supports the development of business integration content for various XML industry standards (ie: [RosettaNet](#) XML for the High Tech Industry, [CIDX](#) for the Chemical Industry and [STAR](#) for the Automotive Industry) on the SAP Exchange Infrastructure (XI) platform of NetWeaver. When he's not solving B2B or SAP integration related issues, you'll find him playing [pinball](#) in his spare time.

Disclaimer & Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.