|  |  |
| --- | --- |
| **Sno** | **Description** |
|  | **Use case Description:**  🡪Service consumer (SOAP UI / NetTool) will send the request in the form of XML to the DataPower.  🡪DataPower will accept it and send response back to the client.  🡪DataPower will transform XML into XSL (eXtensible Stylesheet Language) format and append current Date to the XSL Sheet.  🡪The request will be passed through simply from Datapower and response is sent back.  🡪Make sure that there is Tomcat (any) Server running on Eclipse for any Simple Java Web Application.  🡪Eg. Servlet based aplication  🡪Typical Servlet Code is as below:  @WebServlet("/SimpleXSLCopy")  **public** **class** SimpleXSLCopy **extends** HttpServlet {  **private** **static** **final** **long** *serialVersionUID* = 1L;  **public** SimpleXSLCopy() {  **super**();  }  **protected** **void** doGet(HttpServletRequest request,  HttpServletResponse response) **throws** ServletException, IOException {  System.*out*.println("Received request -----");  StringBuilder buffer = **new** StringBuilder();  BufferedReader reader = request.getReader();  String line;  **while** ((line = reader.readLine()) != **null**) {  buffer.append(line);  }  String data = buffer.toString();  System.*out*.println(data);  System.*out*.println("Received request -----");  String xxmlHelloResponse = "<note xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\" xsi:noNamespaceSchemaLocation=\"Account.xsd\">"  + " <to>Sreenivas Bandaru</to>"  + "<from>Narendra Modi</from>"  + "<heading>Regarding Ministership</heading>"  + " <body>You are confirmed for Home Ministry</body>"  + "</note>";    response.setContentType("application/xml");  response.getWriter().append(xxmlHelloResponse);  }  **protected** **void** doPost(HttpServletRequest request,  HttpServletResponse response) **throws** ServletException, IOException {  doGet(request, response);  }  }  The Test XML to be sent to the Datapower via a client like Nettool is as below:  <?xml version=*"1.0"* encoding=*"UTF-8"*?>  <note xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:noNamespaceSchemaLocation=*"Account.xsd"*>  <to>Sreenivas Bandaru</to>  <from>Narendra Modi</from>  <heading>Regarding Ministership</heading>  <body>You are confirmed</body>  </note>  The |
|  | **Usecase Requirements :** Eclipse-DataPower-Nettoolor SOAP UI. |
|  | **Step 1 :** First of all we need to select XML Firewall from below shown Screen shot. |
|  | **Step 2 :** We need to add one XML Firewall as shown in the figure. |
|  | **Step 3 :** In the Add Advanced option we need to provide with following :  Firewall Name, Firewall Type, Back End, Remote Host, Remote Port, Request and Response Type, Processing Policy.  Remaining are optional. As shown in Screen shot.  🡪Before assigning the Front End Port, we need to check whether the assigning port is free.  🡪We can do that by checking in the TCP Port Status. Screen shot is as below: |
|  | **Step 4 :** Follow following steps.  🡪Select Static Backend in the Firewall Type.  🡪In Back End settings, We have to specify Remote Host & port  🡪Example 192.168.1.128 & 8080.  🡪This setting is for Backend service URL and Port. |
|  |  |
|  | 🡪Processing Policy configuration is vital in XML Firewall implementation.  🡪We need to click the ‘+’ sign to get the next screen. |
|  | **Step 5:**  **🡪** Provide a Policy Name relevant to the implementation as shown above.  🡪Click the New Rule button to create a New Rule as shown in the Screen shot presented below: |
|  | **Step 6 :**  **🡪**Provide an appropriate rule name (Eg. Test\_Rule\_0) as mentioned in the screen shot.  🡪Provide an appropriate direction using the drop down. The drop can be seen in the screen shot for possible options.  🡪The directions are like Client to Server, Server to Client, Both Direction, Error.  🡪Eg. Client to Server direction means, this rule gets invoked when the request is coming from client and passing via DP going to back end server.  🡪In this example, we will create both rules.  🡪1. Client to Server & 2. Server to Client.  🡪Following screen shot is for rule creation from Client to Server.  🡪Here, there is not transformation etc happening.  🡪A Simple pass through for any URL match has been configured. Meaning, no URL based restriction is configured in Client to Server.  🡪For Server to Client, we will implement a transformation, where, the DP will append a date. |
|  | **Step 7 :** |
|  | **Step 8 :**  **🡪**Whenever we create a new rule, we need to mention the matching URL property.  🡪Till the time a matching URL property is not mentioned, the icon is boxed in Yellow Line.  🡪Once an appropriate matching URL is mentioned, the yellow box layer will disappear.  🡪We need to Double click on it to open the configuration window, which is shown as below:  🡪In the screen shot, we need to click on the + symbol to provide a matching rule. |
|  | As soon as we click on ‘+’ sign the new window will be opened. Give a name for it.    🡪Once this is completed, click the buttons 'Apply', 'Apply' & 'Done'. |
|  | Once, this part is done, we need to use 'Results' action icon to configure the Results.  Once Dragged onto the flow, it looks as below:    🡪Double click the Results action icon to configure the Results.  🡪Once configured, the screen shot looks as below: |
|  | 🡪We are done configure rules for the Client to Server.  🡪Below, we will configure rules for the Server to Client.  🡪In the same policy, we need to create a new rule and drag a Transformation Action icon.  🡪Once dragged, it looks as below:    🡪On double click on the icon, we get following screen, that is to be filled as shown.    🡪The Dateappender.xsl will look as below. The XSL has dateappender template as highlighted below.  <?xml version="1.0" encoding="UTF-8"?>  <xsl:stylesheet xmlns:xsl=*"http://www.w3.org/1999/XSL/Transform"*  xmlns:date=*"http://exslt.org/dates-and-times"* xmlns:dpfunc=*"www.datapower.com/extensions/functions"*  xmlns:dp=*"http://www.datapower.com/extensions"* exclude-result-prefixes=*"dp date dpfunc"*  extension-element-prefixes=*"dp"* version=*"1.0"*>  <xsl:template match=*"@\*|node()"*>  <xsl:copy>  <xsl:apply-templates select=*"@\*"* />  <xsl:apply-templates />  </xsl:copy>  </xsl:template>  <xsl:template match=*"to"*>  <xsl:copy-of select=*"."* />  <day>  <xsl:value-of select=*"date:date()"* />  </day>  </xsl:template>  </xsl:stylesheet> |
|  | Once, this is done, apply all the configuration buttons.  🡪If everything is done correctly, the configuration state is shown as 'UP'. |
|  | 🡪In order to test the configuration done on the Datapower, we can use NetTool as a client.  🡪Following URL will invoke the service running on Tomcat via Datapower.  🡪Note that the URL has IP Address of the Datapower, but not the Tomcat Server.  http://192.168.1.41:9003/XMLResponderForDatapower/SimpleXSLCopy  🡪The highlighted part is of Datapower IP & Port. |