Testing Your New Generated SAP NetWeaver Gateway Service

Applies to:
SAP NetWeaver Gateway 2.0 SP02

Summary
In this Article we will focus on how to test the NetWeaver Gateway Service you created using the content generators tool. We will use the Browser for the testing and also a Rest Client tool.

Author: Shiri Levi
Company: SAP
Created on: 24 October 2011

Author Bio
Shiri Levi is part of the Solution Management team for NetWeaver Gateway and Duet Enterprise.
Table of Contents

Introduction: .......................................................................................................................................................... 3
   Testing the Query Operation.......................................................................................................................... 3
   Testing the Read Operation........................................................................................................................... 5
   Testing the Update Operation......................................................................................................................... 6
   Testing the Create Operation......................................................................................................................... 8
Summary............................................................................................................................................................. 9
Copyright........................................................................................................................................................... 10
Introduction:

So you have successfully created an SAP service using the generators tool of SAP NetWeaver Gateway. Well, how can you test your service, making sure that it works?

In this article, we will focus on testing an already generated SAP service.

For our purpose, we have generated a service called ZIntegration_EPM_SCR, and we will test the following four operations:

- Query
- Read
- Update
- Create

For each operation, we will perform some manual steps that require the use of a REST Client application and a Web browser.

The prerequisites for this article are as follows:

1. A data model (we created a data model with four operations, ZIntegration_EPM_SCR)
2. Having consumption model created from our data model (ZIntegration_EPM_SCR_0001)
3. Having REST Client application installed (in this article we will use the Firefox REST Client)

Testing the Query Operation

We will test the Query operation in our service from the ABAP Workbench of the SAP NetWeaver Gateway system, using transaction SE80. From the workbench, we will run the data model in a Web browser; as a result we should be able to see a list of backend objects in the browser

1. From the Repository Browser of the ABAP Workbench, we locate our consumption model called, ZIntegration_EPM_SCR_0001_PS under GW Data Model.

2. Open Related GW Consumption under Object Name, and double click the service document to open the Consumption Model Header screen and click the XML button.

The browser will request for user credentials. Provide the user credentials for the backend system from which the business data will be coming.

The browser opens to display the XML in the service document as shown below:
3. Select the root collection of your service model, **zintegration_epm_scrCollection**, (you can recognize it by the name, it will have the same name as our data model), and insert before **?sap-client** in the URL as follows and then press **Enter**:

```
http://<host_name>:<port_number>/sap/opu/sdata/sap/zintegration_epm_ScrCollection?sap-client=100&$format=xml
```

We should get a list of backend objects, in our scenario, which is a list of products from the EPM scenario.

Below is an example of the results of a Query operation

```xml
<atom:entry href="/sap/opu/sdata/sap/zintegration_epm_scrCollection" title=""/>
<atom:entry href="/sap/opu/sdata/sap/zintegration_epm_scrCollection" title=""/>
<atom:entry href="/sap/opu/sdata/sap/zintegration_epm_scrCollection" title=""/>
<atom:entry href="/sap/opu/sdata/sap/zintegration_epm_scrCollection" title=""/>
```
Testing Your New Generated SAP NetWeaver Gateway Service

We will test the Read operation using the Web browser to obtain the details of a specific backend object.

We will select a specific collection from the results of the query and insert it in the URL.

1. From the query results in the browser, copy the value of the href:

```
zintegration_epm_scrCollection(value='HT1000',scheme_id='ZINTEGRATION_EPM_SCR',scheme_agency_id='G2R_TRUSTED_800')/product_classic_dynpro_0200_r
```

2. Paste it into the URL before ?sap-client as follows and press Enter:

```
http://<host_name>:<port_number>/sap/opu/sdata/sap/ZINTEGRATION_EPM_SCR/zintegration_epm_scrCollection(value='HT-1000',scheme_id='ZINTEGRATION_EPM_SCR',scheme_agency_id='G2R_TRUSTED_800')/product_classic_dynpro_0200_r
```

We should get the business data for a specific backend object, a product ID 'HT-1000' from the EPM scenario.

Below is an example of the results in XML:

```
<entry width="150" encoding="WHT-4">
  <id>http://www.sap.com/opu/sdata/sap/ZINTEGRATION_EPM_SCR?sap-client=100</id>
  <title>product_classic_dynpro_0200_r</title>
  <updated>2011-10-05T13:43:45Z</updated>
  <author>
    <name>John Doe</name>
  </author>
  <content type="application/vnd.sap.com/Protocols/ContentText" xml:base="http://www.sap.com/Protocols/ContentText">
    <m:properties>
      <m:product_classic_dynpro_0200_r />
    </m:properties>
  </content>
</entry>
```

For the testing the Update operation, we will take information from the results of the Read operation from the Web browser.

Copy the information between the tag, `<m:properties>` , and the tag,`</m:properties>`, and paste it into a Notepad. Later, we will use it. 😊
Testing the Update Operation

We will test the Update operation using a REST client application, and the information retrieved from the results of the Read operation.

We will use the following methods in the REST client:

- Get = Read
- Put = Update
- Post = Create

First, we test the client by running a Get method, just to see how it works.

1. Select "Get" from the list of methods.

2. Copy and paste the URL from which we test the Read operation in the text box next to the Method button, and click the Send button. The result should be as follows:

Open the "format XML" tab to see the actual fields derived from the backend system:

To test our Update operation:

1. Select the PUT method, and click the "Add Request Header" button, and then insert the following parameters into the Request Header.
   - "Name" = X-Requested-With
   - "Value" = XMLHttpRequest

2. We assemble the request body, using our information from the Notepad. The request template is as follows:
3. Paste the entire request into the request body:

```
<atom:entry xmlns:atom="http://www.w3.org/2005/Atom">
  <atom:title>Update Test</atom:title>
  <atom:content type="application/xml">
      Paste the information from the results of the Read operation here
    </m:properties>
  </atom:content>
</atom:entry>
```

4. In Request Body, we select the parameters we want to update, product ID “HT-1000” and the attribute, Weight, for example we want to increase the current weight 4.600 to 4.900.
5. Click "Send" button.
   The results of the request should be "No Content" and green in color.

Using the Get method again, we check if the return value in the formatted XML tab contains the changes we have just made. Alternatively, just logon to the Backend system and check if it was updated.

**Testing the Create Operation**

We will test the Create operation using our REST client application and the information retrieved from the results of both the Read and the Query operations.

1. Select the "Post" method from the REST Client methods, and provide the URL of the results of the Query operation.

2. Add the Request Header data using the procedure in the Update operation above.

3. In the Request Body (as created in the Update operation test above) we need to provide the object ID if we want to test the Create operation, so you need to check in the backend system which product ID is free and provide it here.

   In our example we checked and found that there is no product ID in the system with the ID HT-4000, so we will create a new one with this ID:

   ```xml
   <d:value>[value to delete and replace with new BACKEND ObjectID]</d:value>
   
   For example, `<d:value>HT-4000</d:value>`
   
   Sometimes the ID appears several times in the Request Body, so verify that you replace them all.
Since we are creating a new backend object (in this example a new product ID), we need to provide all the mandatory information required to create it, otherwise we will not be able to create new Backend object.

4. Click the **Send** button

The result should be "Created" and a new backend object is created in the backend system.

**Summary**

In summary, it is possible to verify the end-to-end (E2E) functionality of your SAP service; and now you can test it before using it from any consumer environment of your choice, such as, Web, device, and many more.
Copyright

© Copyright 2012 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors. Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

Java is a registered trademark of Oracle Corporation.

JavaScript is a registered trademark of Oracle Corporation, used under license for technology invented and implemented by Netscape.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP Business ByDesign, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects S.A. in the United States and in other countries. Business Objects is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies (“SAP Group”) for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.