How to test Sage X3 PU9 (and above) Web Services using SoapUI tool with OAuth2 Authentication

Disclaimer

This document is provided "as is" and is for your guidance and educational purposes only. It does not replace the Online documentation, nor is any warranty expressed nor implied for the steps described below.

NOTE: Sage do not specifically recommend nor provide any support or warranty for the SoapUI tool

Document Information

Author: Mike Shaw, Sage UK X3 Support Team

Introduction

This document is a walk through as to how to send a REST request to an X3 instance, using 3rd party application "SoapUI", when you wish to use OAuth2 as the authentication mechanism (rather than basic authentication)

The assumptions used in this example are:

a) SoapUI tool is already installed and can access the Sage X3 Web Server, either over the network or is running directly on the Web Server itself and can also access your OAuth2 provider. In this example we are using Microsoft as our provider.
b) Web Services are already configured and running within X3 and you have correct license to allow Web Services to run

The documentation for setting up SoapUI with OAuth2 is available from their web site at https://www.soapui.org/oauth2/oauth2-overview.html

NOTE : SoapUI only supports OAuth2 authentication for REST requests

Prepare your Sage X3 environment

There are 2 steps you need to do to setup your Sage X3 Syracuse (Web) Server

1. Update Syracuse configuration
   Navigate to the Syracuse\bin directory, for example C:\Sage\SAFEX3\X3PU9SYRA\syracuse\bin
   Copy the existing “nodelocal.js” as a backup
   Edit this file and add “bearer” as an authentication method

   For example, change the following line

   ```
   "auth": [ "basic", "oauth2"]
   ```

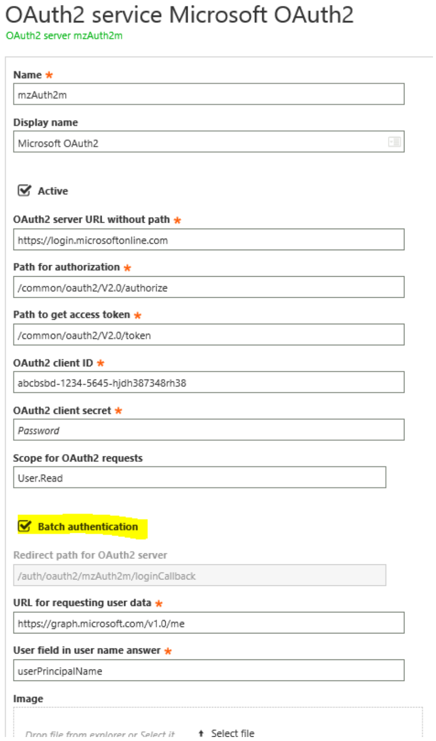
   To read

   ```
   "auth": [ "basic", "oauth2", "bearer"]
   ```

2. Configure the OAuth2 Server for “Batch authentication”

Login to X3 and navigate to Administration→ Settings→ Authentication→ Oauth2 Servers

Check the “Batch authentication” checkbox and save the change



OAuth2 service Microsoft OAuth2

OAuth2 server mzAuth2m

Name *
mzAuth2m

Display name
Microsoft OAuth2

☑ Active

OAuth2 server URL without path *
https://login.microsoftonline.com

Path for authorization *
/common/oauth2/V2.0/authorize

Path to get access token *
/common/oauth2/V2.0/token

OAuth2 client ID *
abcbsbd-1234-5645-hjdh387348rh38

OAuth2 client secret *
Password

Scope for OAuth2 requests
User.Read

☑ Batch authentication

Redirect path for OAuth2 server
/auth/oauth2/mzAuth2m/loginCallback

URL for requesting user data *
https://graph.microsoft.com/v1.0/me

User field in user name answer *
userPrincipalName

Image
Drop file from explorer or Select it

Then restart your Syracuse service to pick up these changes

NOTE: If you have two or more OAuth2 providers defined with batch authentication ticked, you need to add a parameter to the URL :

oauth2=name_of_the_provider

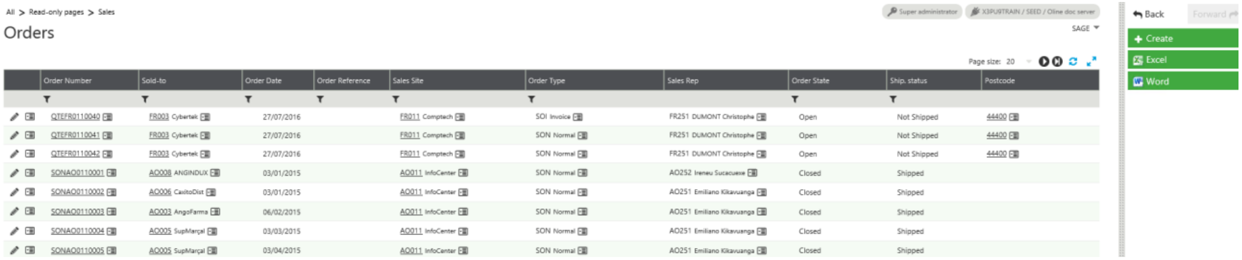
where “name_of_the_provider” is the name used in the redirect URL

Create your test case in Sage X3

Whilst logged into X3, create a test case which we will then emulate in SoapUI later on

For this example, I will use Sales orders

a. Navigate to Read-only pages→ Sales→ Sales orders



All > Read-only pages > Sales

Orders

Page size: 20

Order Number	Sold-to	Order Date	Order Reference	Sales Site	Order Type	Sales Rep	Order State	Ship. status	Postcode
QTEFR0110040	FR003 Cybertek	27/07/2016		FR011 Comptech	SOI Invoice	FR251 DUMONT Christophe	Open	Not Shipped	44400
QTEFR0110041	FR003 Cybertek	27/07/2016		FR011 Comptech	SON Normal	FR251 DUMONT Christophe	Open	Not Shipped	44400
QTEFR0110042	FR003 Cybertek	27/07/2016		FR011 Comptech	SON Normal	FR251 DUMONT Christophe	Open	Not Shipped	44400
SONAO0110001	AO008 ANGINDUX	03/01/2015		AO011 InfoCenter	SON Normal	AO252 Ireneu Sucacuexe	Closed	Shipped	
SONAO0110002	AO006 CaxitoDist	03/01/2015		AO011 InfoCenter	SON Normal	AO251 Emiliano Kikavuanga	Closed	Shipped	
SONAO0110003	AO003 AngoFarma	06/02/2015		AO011 InfoCenter	SON Normal	AO251 Emiliano Kikavuanga	Closed	Shipped	
SONAO0110004	AO005 SupMarçal	03/03/2015		AO011 InfoCenter	SON Normal	AO251 Emiliano Kikavuanga	Closed	Shipped	
SONAO0110005	AO005 SupMarçal	03/04/2015		AO011 InfoCenter	SON Normal	AO251 Emiliano Kikavuanga	Closed	Shipped	

Make note of the URL in the browser, for example https://server.example.com/syracuse-main/html/main.html?url=%2Fsdata%2Fx3%2Ferp%2FX3PU9TRAIN_SEED_ONLINEDOC%2FSORDER%3Frepresentation%3DSORDER.%24query

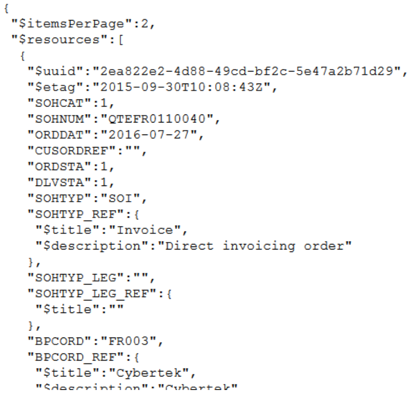
Take the “url” parameter from this URL, convert to human readable form (optional) and add back in the protocol, host.domain and port number (as appropriate) to get to the URL below:

https://server.example.com/sdata/x3/erp/X3PU9TRAIN_SEED_ONLINEDOC/SORDER?representation=SORDER.$query&count=2

The “&count=2” is an optional parameter, allowing you to specify the number of records to be returned

We can now take this URL and paste it into our browser window to test it works OK.   We are expecting to see some JSON output from this URL

NOTE: Internet Explorer might insist you download the output as a file by default, but Firefox seems happy to display the output in the browser window



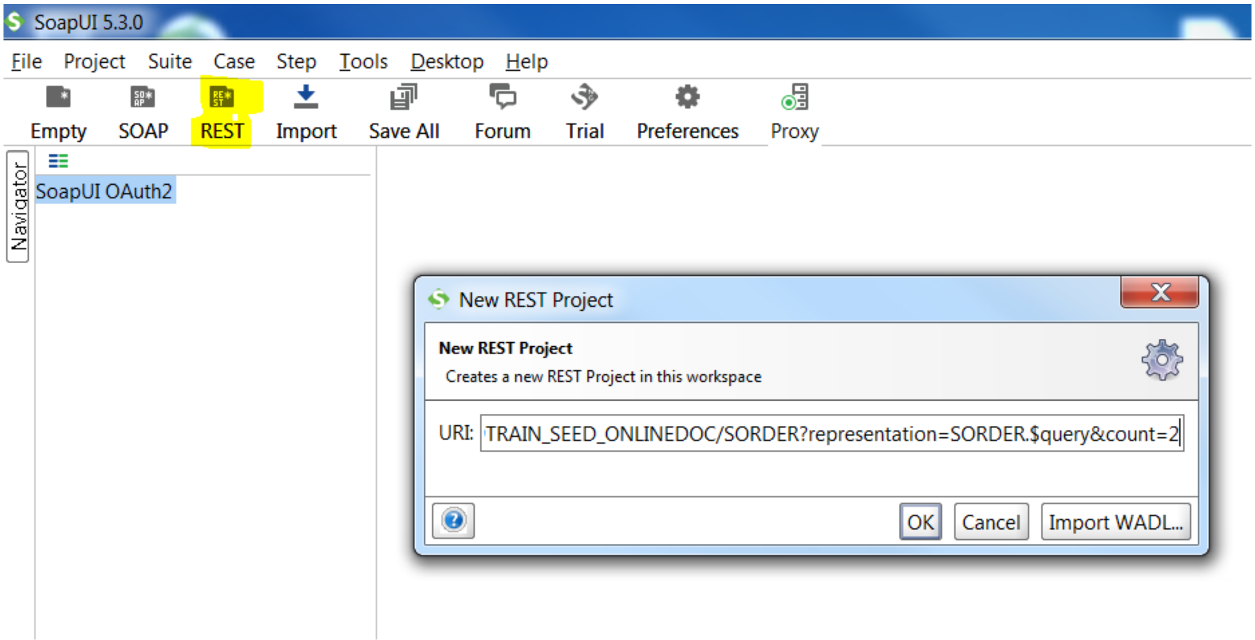
```
{
 "$itemsPerPage":2,
 "$resources":[
  {
   "$uuid":"2ea822e2-4d88-49cd-bf2c-5e47a2b71d29",
   "$etag":"2015-09-30T10:08:43Z",
   "SOHCAT":1,
   "SOHNUM":"QTEFR0110040",
   "ORDDAT":"2016-07-27",
   "CUSORDREF":"",
   "ORDSTA":1,
   "DLVSTA":1,
   "SOHTYP":"SOI",
   "SOHTYP_REF":{
    "$title":"Invoice",
    "$description":"Direct invoicing order"
   },
   "SOHTYP_LEG":"",
   "SOHTYP_LEG_REF":{
    "$title":""
   },
   "BPCORD":"FR003",
   "BPCORD_REF":{
    "$title":"Cybertek",
    "$description":"Cybertek"
```

Once we see this output, we can now go to SoapUI tool to replicate the same request

Launch SOAPUI tool

1. Create new REST project

Enter the URL we used earlier when prompted https://server.example.com/sdata/x3/erp/X3PU9TRAIN_SEED_ONLINEDOC/SORDER?representation=SORDER.$query&count=2



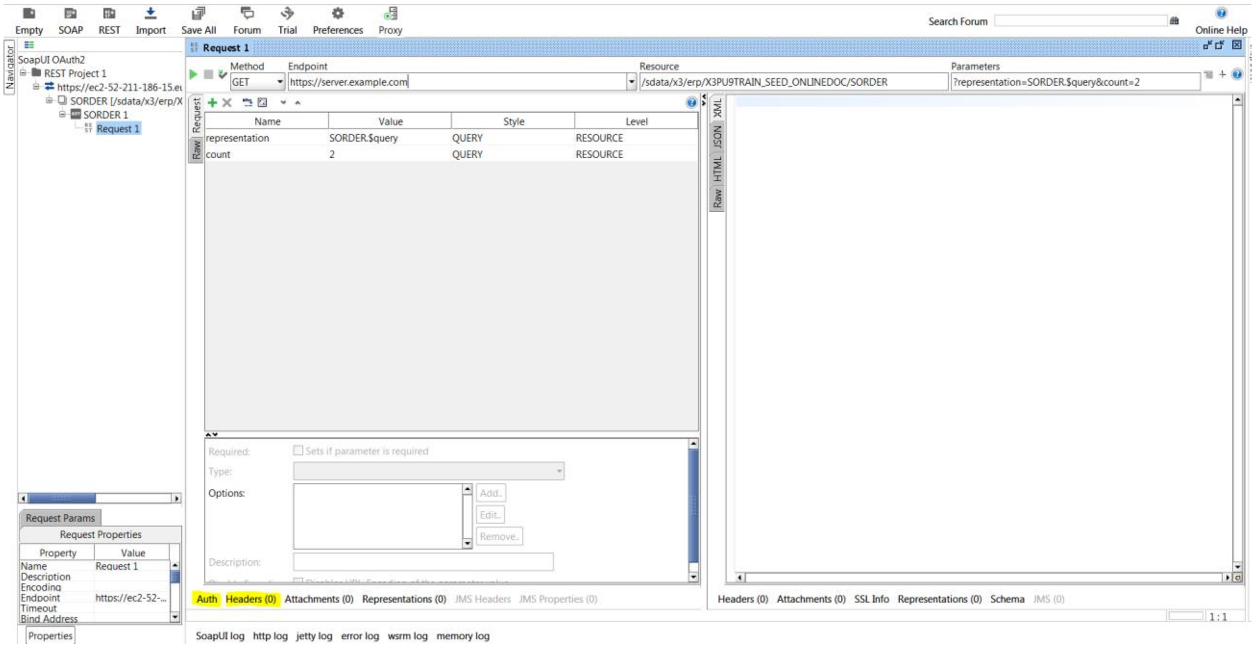
SoapUI 5.3.0
File Project Suite Case Step Tools Desktop Help
Empty SOAP REST Import Save All Forum Trial Preferences Proxy
Navigator
SoapUI OAuth2
New REST Project
OK Cancel Import WADL...

New REST Project

Creates a new REST Project in this workspace

URI: TRAIN_SEED_ONLINEDOC/SORDER?representation=SORDER.$query&count=2

Once you click OK, you see the following screen



Request 1

Method: GET
Endpoint: https://server.example.com
Resource: /sdata/x3/erp/X3PU9TRAIN_SEED_ONLINEDOC/SORDER
Parameters: ?representation=SORDER.$query&count=2

Name	Value	Style	Level
representation	SORDER.$query	QUERY	RESOURCE
count	2	QUERY	RESOURCE

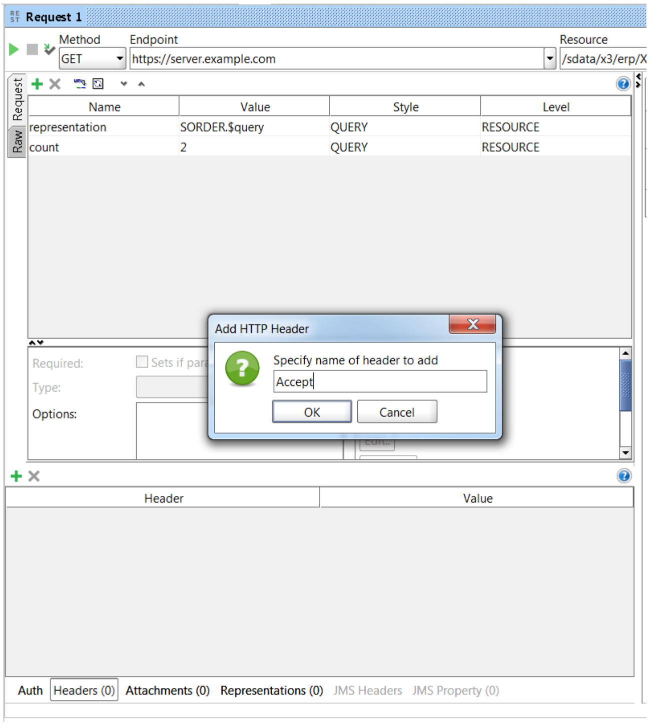
Required: □ Sets if parameter is required
Type:
Options: Add.. Edit.. Remove..
Description:

Property	Value
Name	Request 1
Description	
Encoding	
Endpoint	https://ec2-52-...
Timeout	
Bind Address	

We’ll do the easier part first, which is to add a header "Accept" with value "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8"

Click the “Headers (0)” at the bottom of the screen, then the green + sign

Enter “Accept” then click OK



Request 1

Method: GET
Endpoint: https://server.example.com
Resource: /sdata/x3/erp/X

Name	Value	Style	Level
representation	SORDER.$query	QUERY	RESOURCE
count	2	QUERY	RESOURCE

Required: □ Sets if par
Type:
Options:

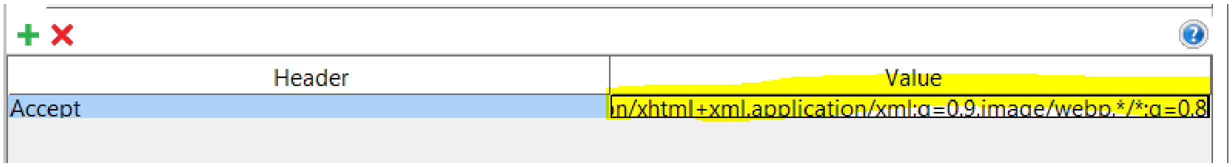
Add HTTP Header

Specify name of header to add

Accept

Header	Value

Enter the value as shown

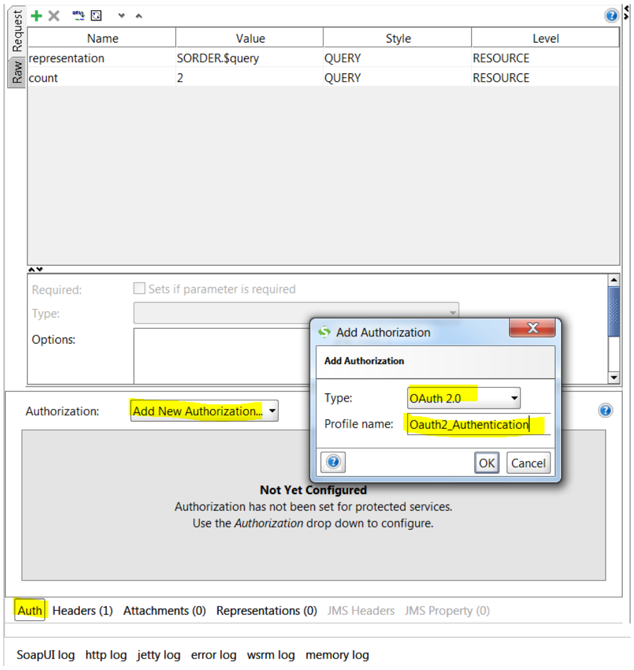


Header	Value
Accept	n/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

Next we setup authentication for our request

You can setup multiple authentication types and switch between them if you wish (for the same request). For example, you could first setup BASIC authentication to test the request, then add another authorization type and configure OAuth2 – in this example however, we will go straight into OAuth2

Click the "Auth" at the bottom of the screen, then select "Add new authentication"
Select "OAuth 2.0" from the list of values and give the profile a name (optional)



Name	Value	Style	Level
representation	SORDER.$query	QUERY	RESOURCE
count	2	QUERY	RESOURCE

Required: □ Sets if parameter is required
Type:
Options:

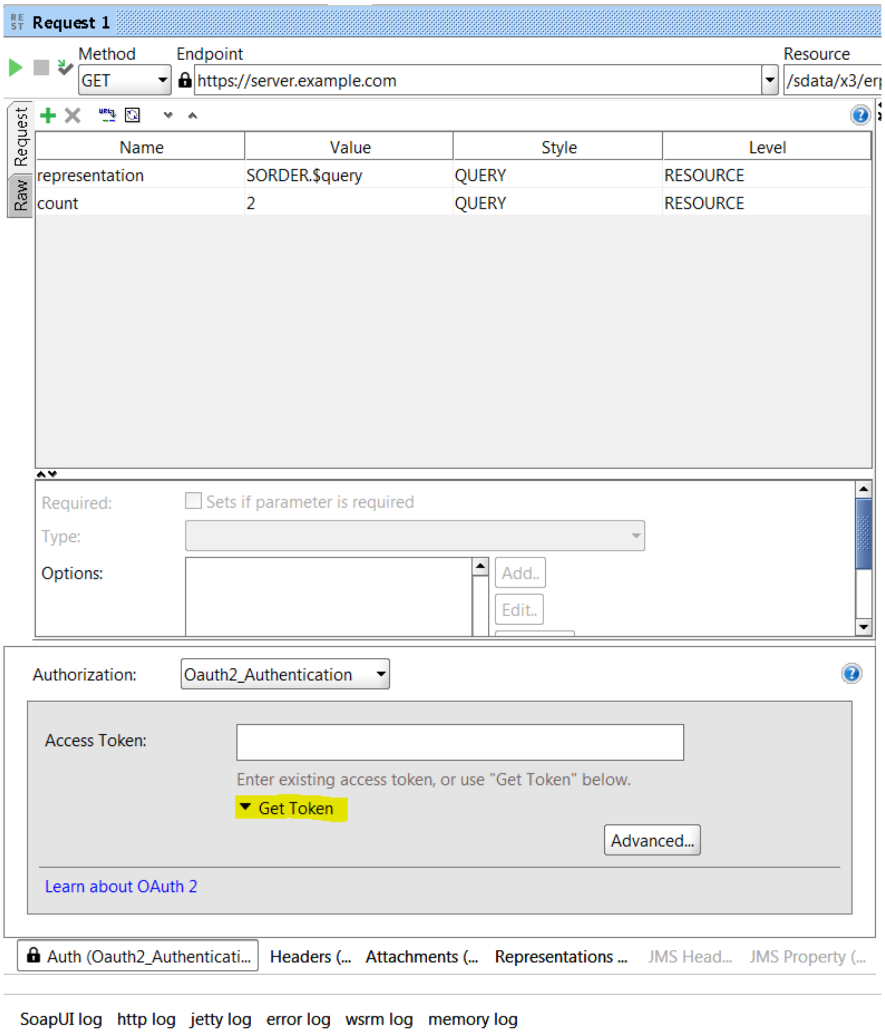
Authorization: Add New Authorization...

Add Authorization

Type: OAuth 2.0
Profile name: Oauth2_Authentication

Not Yet Configured
Authorization has not been set for protected services.
Use the *Authorization* drop down to configure.

You will then see the option to "Get Token" which you need to select



Request 1

Method: GET

Endpoint: https://server.example.com

Resource: /sdata/x3/erp

Name	Value	Style	Level
representation	SORDER.$query	QUERY	RESOURCE
count	2	QUERY	RESOURCE

Required: □ Sets if parameter is required

Type:

Options:

Authorization: Oauth2_Authentication

Access Token:

Enter existing access token, or use "Get Token" below.

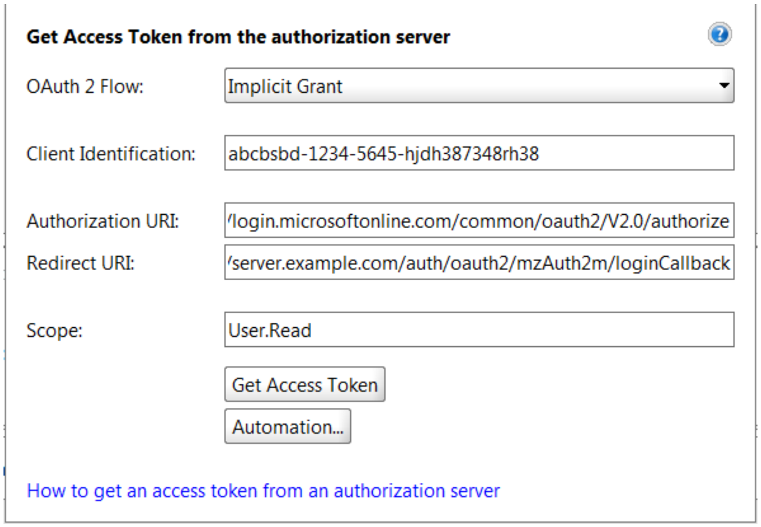
▼ Get Token

At this point you need to fill out the details in order to validate against your OAuth2 provider.   This information you will gather from either your X3 OAuth2 server setup, or from your OAuth2 providers setup (both of which has already been done when you integrated X3 with OAuth2)

Select "Implicit Grant" and enter the following

- "Client Identification" is the "OAuth2 client ID" from the OAuth2 server setup in X3
- "Authorization URI" is the "OAuth2 server URL without path" plus the "Path for authorization" from the OAuth2 server setup in X3
- "Redirect URI"   : https://host.domain:port/auth/oauth2/mzAuth2m/loginCallback which was entered in your OAuth2 Application setup.   This is your Sage X3 syracuse hosts "host.domain:port"
- "Scope" is the value "User.Read", same as the OAuth2 server setup in X3

Your completed screen should look something like below:



Get Access Token from the authorization server

OAuth 2 Flow: Implicit Grant

Client Identification: abcbsbd-1234-5645-hjdh387348rh38

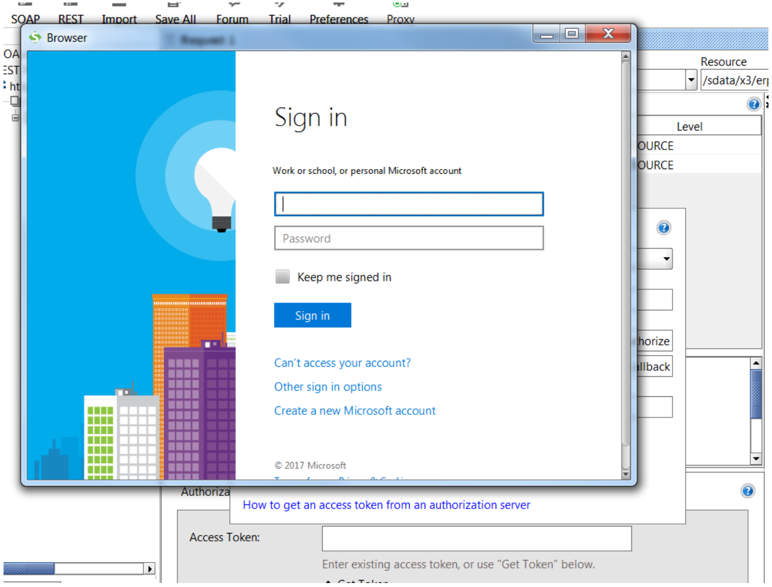
Authorization URI: /login.microsoftonline.com/common/oauth2/V2.0/authorize

Redirect URI: /server.example.com/auth/oauth2/mzAuth2m/loginCallback

Scope: User.Read

The next step is to authenticate your SoapUI session, by clicking the “Get Access Token” button

This provides a screen for you to login to your (in this case) Microsoft account



Sign in

Work or school, or personal Microsoft account

Password

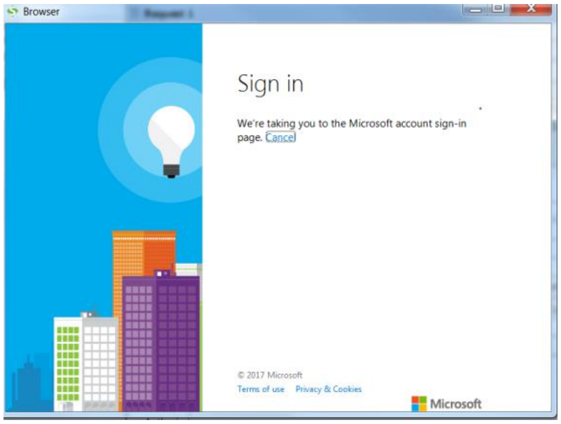
□ Keep me signed in

Access Token:

Enter existing access token, or use "Get Token" below.

© 2017 Microsoft

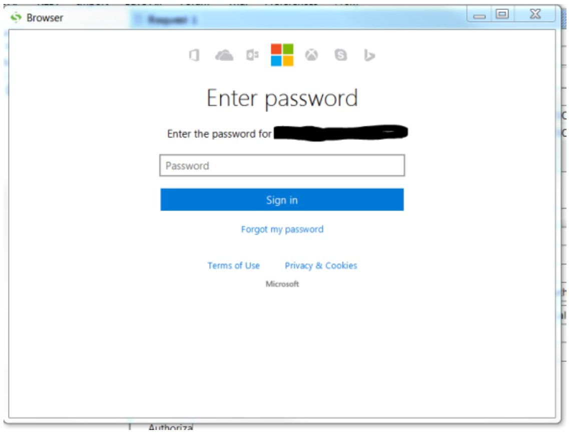
Note that when you fill in the email address and tab to the password field, the Sage X3 callback URL will be executed before you are then presented with the password entry screen. If you get an error at this point, it is most likely a configuration or network issue. For example, the callback URL is wrong, or cannot be accessed from the OAuth2 provider server.



Sign in

We're taking you to the Microsoft account sign-in page. Cancel

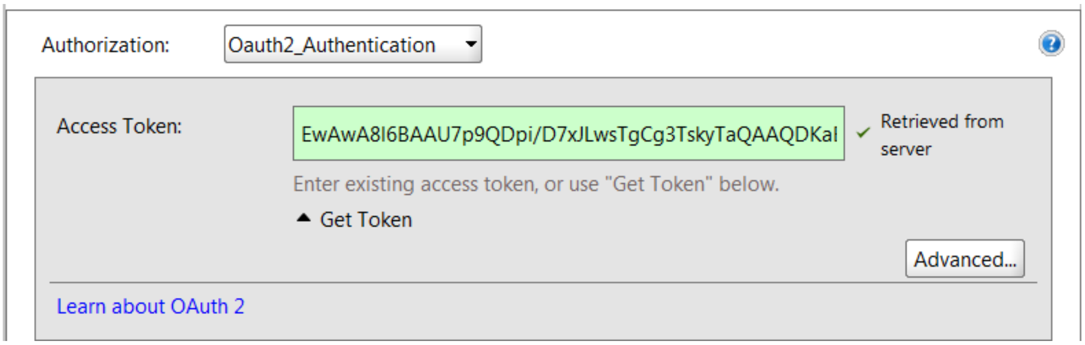
© 2017 Microsoft



Enter password

Enter the password for [redacted]

Once successful, you will see your Access Token in SoapUI, coloured Green if it's valid.

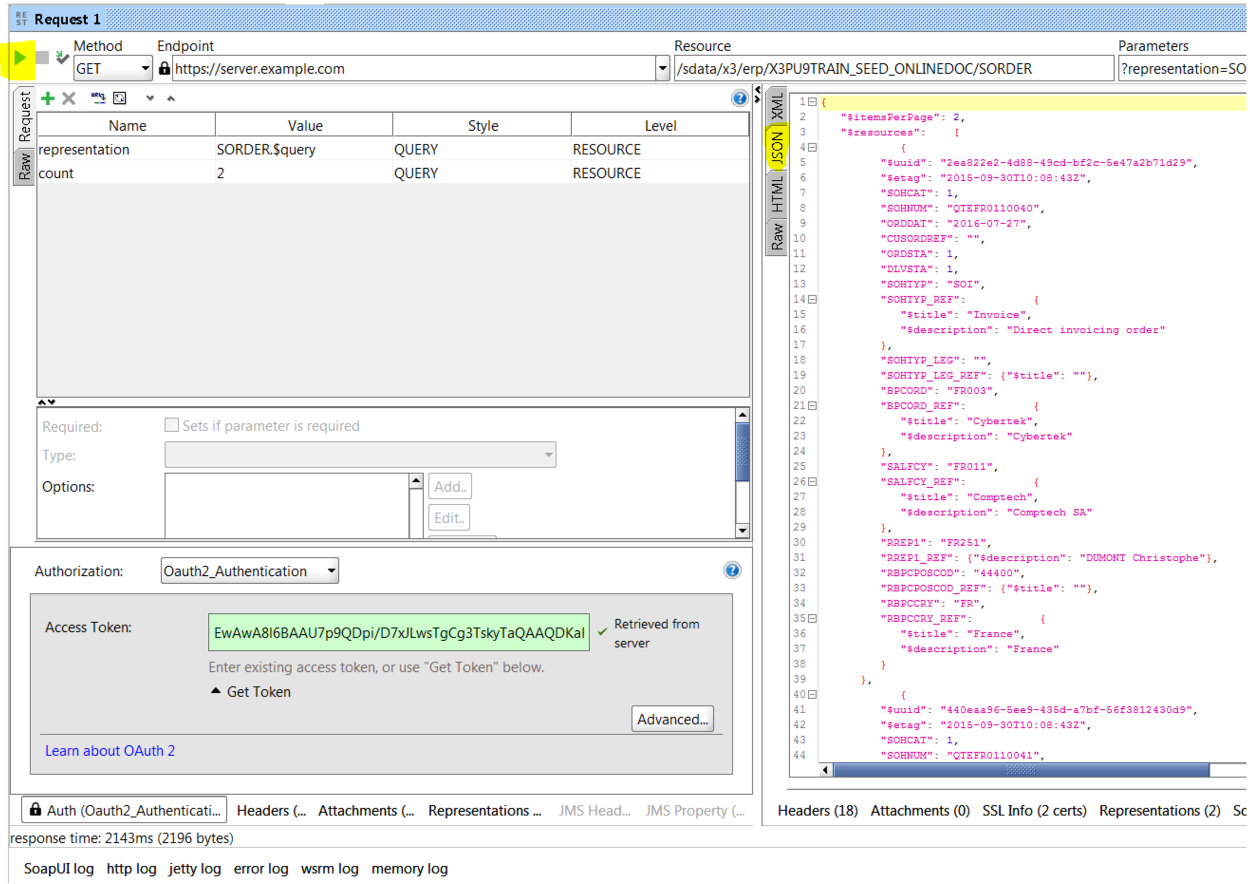


Authorization: Oauth2_Authentication

Access Token: EwAwA8I6BAAU7p9QDpi/D7xJLwsTgCg3TskyTaQAAQDKal ✓ Retrieved from server

Enter existing access token, or use "Get Token" below.

Finally, we can now submit our REST request, using the green run button. Once completed, click the "JSON" tab in the response window. If all is well, you will see the JSON output the same as shown in the browser window earlier



Method: GET
Endpoint: https://server.example.com
Resource: /sdata/x3/erp/X3PU9TRAIN_SEED_ONLINEDOC/SORDER
Parameters: ?representation=SO

Name	Value	Style	Level
representation	SORDER.$query	QUERY	RESOURCE
count	2	QUERY	RESOURCE

Required: □ Sets if parameter is required
Type:
Options:

Authorization: Oauth2_Authentication

Access Token: EwAwA8I6BAAU7p9QDpi/D7xJLwsTgCg3TskyTaQAAQDKal ✓ Retrieved from server

Enter existing access token, or use "Get Token" below.

response time: 2143ms (2196 bytes)

```
{
   "$itemsPerPage": 2,
   "$resources":    [
            {
         "$uuid": "2ea822e2-4d88-49cd-bf2c-5e47a2b71d29",
         "$etag": "2015-09-30T10:08:43Z",
         "SOHCAT": 1,
         "SOHNUM": "QTEFR0110040",
         "ORDDAT": "2016-07-27",
         "CUSORDREF": "",
         "ORDSTA": 1,
         "DLVSTA": 1,
         "SOHTYP": "SOI",
         "SOHTYP_REF":          {
            "$title": "Invoice",
            "$description": "Direct invoicing order"
         },
         "SOHTYP_LEG": "",
         "SOHTYP_LEG_REF": {"$title": ""},
         "BPCORD": "FR003",
         "BPCORD_REF":          {
            "$title": "Cybertek",
            "$description": "Cybertek"
         },
         "SALFCY": "FR011",
         "SALFCY_REF":          {
            "$title": "Comptech",
            "$description": "Comptech SA"
         },
         "RREP1": "FR251",
         "RREP1_REF": {"$description": "DUMONT Christophe"},
         "RBPCPOSCOD": "44400",
         "RBPCPOSCOD_REF": {"$title": ""},
         "RBPCCRY": "FR",
         "RBPCCRY_REF":          {
            "$title": "France",
            "$description": "France"
         }
      },
            {
         "$uuid": "440eaa96-5ee9-435d-a7bf-56f3812430d9",
         "$etag": "2015-09-30T10:08:43Z",
         "SOHCAT": 1,
         "SOHNUM": "QTEFR0110041",
```

If there are errors at this stage, check the "Raw" tabs for both the request and response for clues, plus also review the SoapUI log

Conclusion

Hopefully this worked example document has provided you with additional clarifications to accompany the Online help, demonstrating how to interface an External Web Service with Sage X3 when using OAuth2 Authentication