How to integrate Denodo with SharePoint Online

Revision 20221229

CONTENTS

This document describes all the possible ways to access SharePoint Online from Denodo Virtual DataPort:

- The fastest way that can be enough for many users is to use the **Denodo Templates for SharePoint**. These templates allow accessing information like Lists, Items, File and Folder Metadata of any SharePoint site. If you want to use the Denodo templates for SharePoint take a look at the Denodo Templates for SharePoint - Quick User Guide. Note that, even using the templates, you have to register and configure the app in Azure as explained in this document. In case you do not want to use the templates or require information not available through them, check the other options.
- **OData2 Custom Wrapper**: to access SharePoint OData entities.
- **Excel**, **JSON**, **XML** and **Delimited File Data Sources**: to access SharePoint's REST API to retrieve resources like Files, Lists or Folders.
- Also, you can connect to SharePoint online through **RSS.**

The following table summarizes the possible ways to access SharePoint:

NEED	CONNECTION METHOD	AUTHENTICATION
Access metadata of Files, Folders, Lists or Items; and basic OData entities on any	Denodo Templates for Sharepoint	OAuth 2.0

Sharepoint site.		
Access other OData entities.	OData2 Custom Wrapper	
Retrieve metadata of Files, Folders, Lists or Items (without using the Denodo Templates for SharePoint).	JSON / XML data sources	OAuth 2.0
Read Excel files.	Excel data source	OAuth 2.0
Read CSV files.	Delimited File data source	OAuth 2.0
Read Sharepoint Lists.	RSS	OAuth 2.0

***NOTE**: NTLM authentication is only an option for SharePoint Server, not for SharePoint Online.

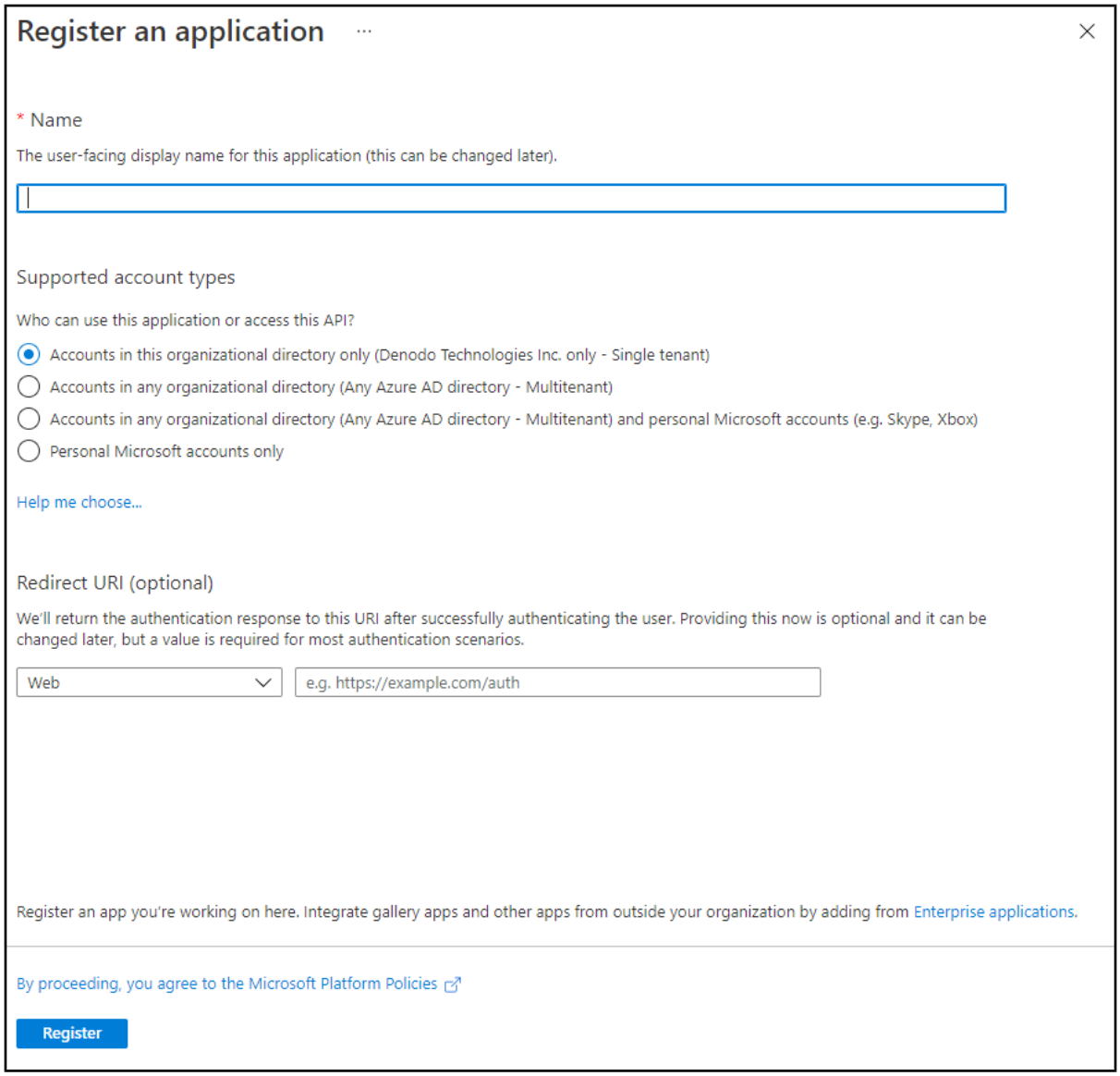
In all cases it is necessary to register and configure an app in Azure as described below.

1 CONFIGURING THE APPLICATION IN AZURE

First of all, it is necessary to register an App, provide the correct API permissions and generate a client secret in the Azure Portal in order to allow the app (in this case Denodo) to access SharePoint.

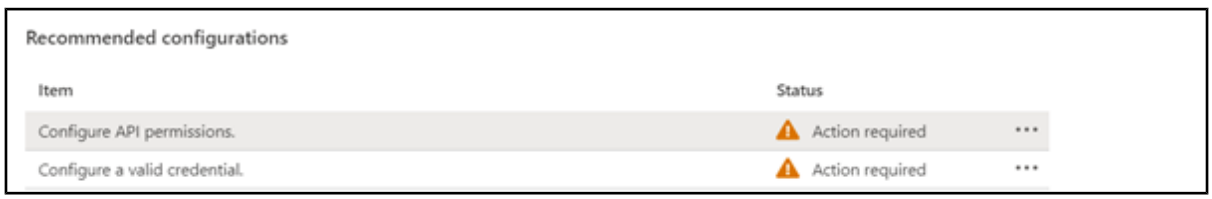
In order to register an app in Azure, follow these steps:

1. Login to the Azure portal
2. Navigate to https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps
3. Press the button `New Registration`
4. Give a name for the app
5. Fill in the `Support Account Types` field. The type of the application support depends on your requirement. For the purpose of this documentation we are using `Accounts in this organizational directory only (Default Directory only - Single tenant)`.
6. Enter `Redirect URL` value. Due to a SharePoint limitation, the value you set at this time will be the only one you can use as a redirect URL for this app. Azure supports adding more redirect URLs to an app but Sharepoint only supports one.
   Although Azure allows its later modification, this modification will not be synchronized with SharePoint and will not work. The default value is http://localhost:9090/oauth/2.0/redirectURL.jsp. Note that http only can be used with localhost, if you have HTTPS enabled in your Denodo Virtual DataPort, then use your hostname instead of localhost: https://<hostname>:9443/oauth/2.0/redirectURL.jsp.
7. Press the button "`Register`".



Once the app is registered we will need to grant permissions for clients to access SharePoint Online using the app:

- In the newly registered application, click on `Integration assistant` (on the left side menu) and select `Daemon` for `What application types are you building` and select `Evaluate my registration`.
- Once the evaluation is done, the Recommended configurations are shown.

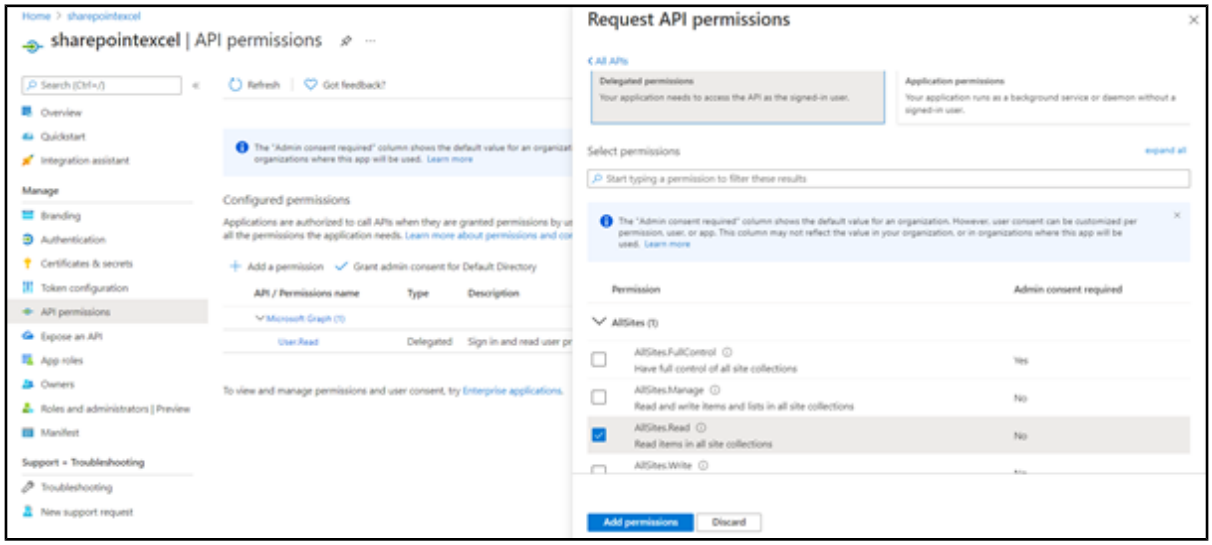


There are 2 needed configurations that have to be performed to allow the Denodo

Platform to access the data in SharePoint: API permissions and client secret generation.

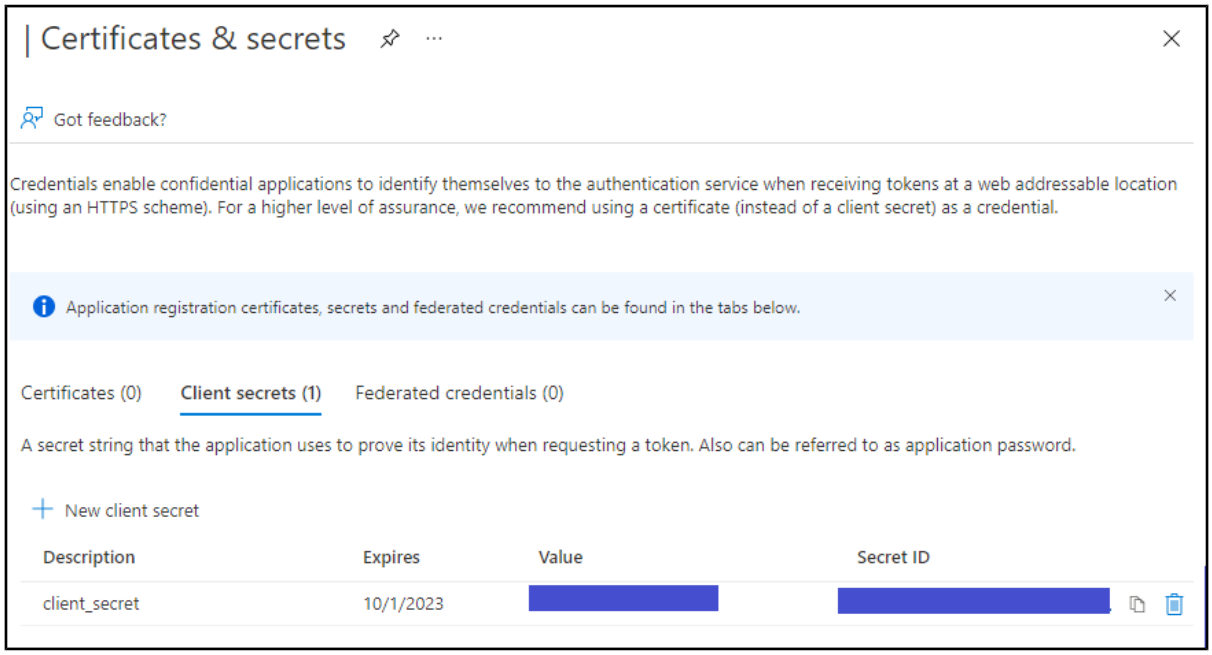
In order to configure API permissions follow the following steps:

- Click on the ••• icon on `Configure API permissions` and select `Go to` page.
- Select `Add Permissions`. In the Request API permissions pop-up select SharePoint.
- Select `Delegated Permissions` and select `AllSites.Read` as the permissions.



After the API permissions are set we have to generate a new client secret key. In order to generate the key:

- In the `Integration assistance` section, select `Configure a valid credential`.
- In the `Certificates & secrets` page select the option `New client secret`.
- After generating the Client secret, copy the **`Value`**.



After having followed those sections, you should have gathered these values:

- **`Application (client) ID`** which will be the **`Client Id`** in the data sources configuration.
- **`Value of the Client Secret`** from step “Client secret key generation”
- The **`Directory (tenant) ID`** which will be the **`tenant_id`** needed later.

2 CONNECTING TO SHAREPOINT USING ODATA

2.1 OBTAIN THE OAUTH TOKENS

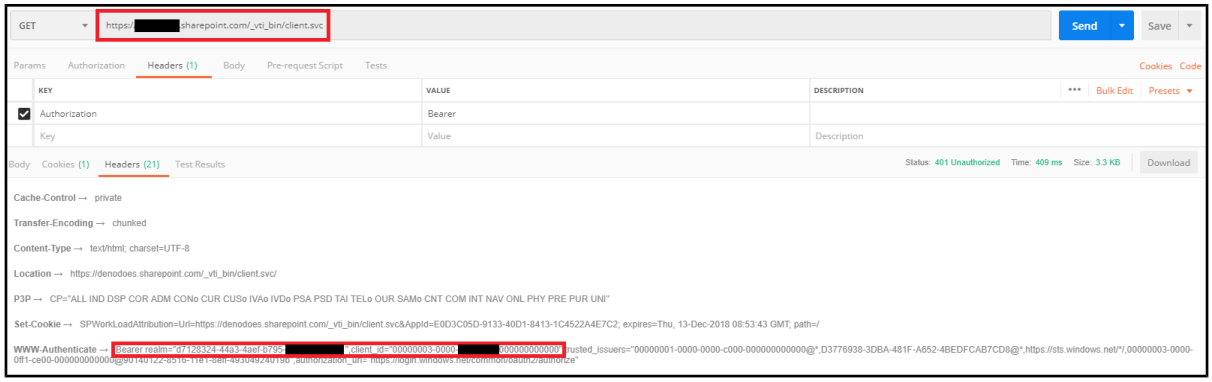
NOTE: If you are not going to use the OAuth authentication you can skip this step.

As you have obtained the client id, client secret and tenant id, you need to obtain the access token and refresh token by using those values.

2.1.1 Get the Realm of the site

Realm is a constant GUID for a site. In order to retrieve it, a tool such as Postman can be used. To obtain the Realm, it is necessary to carry out the following steps:

- Make a GET request like this:
    - `https://<tenant_name>.sharepoint.com/_vti_bin/ListData.svc`
    - Header:
        - `Authorization: Bearer`
- Get the `Bearer realm` component from the `WWW-Authenticate` response header and save it. This value is the same as the tenant id.
- Get the `client_id` component from the `WWW-Authenticate` response header and save it. This value is what later we will call Audience Principal ID.



NOTE: If you are using cookies you might not get back the `WWW-Authenticate` header with the Bearer realm. Delete the cookies in the browser or use the Postman Interceptor to avoid this issue.

2.1.2 Get the authorization code

Construct an authorization url as follows:

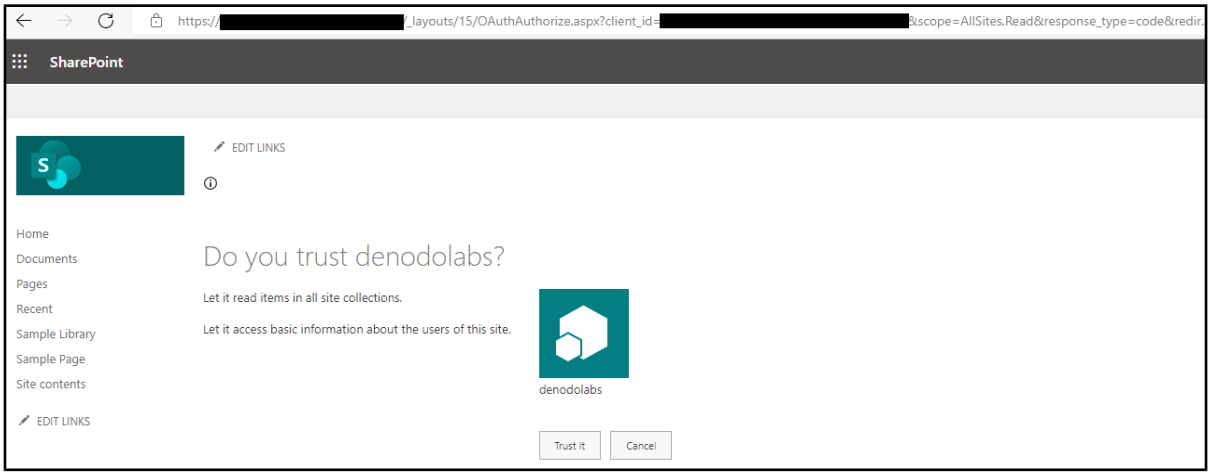
```
https://<tenant_name>.sharepoint.com/_layouts/15/OAuthAuthorize.aspx?
client_id=<client_id>&scope=AllSites.Read&response_type=code&redirect_uri
=<redirect_uri>
```

Change the parameters of the URL to fit your connection data. For the `client_id` use the **`Application (client) ID`** from the application registration in the Azure portal (this is different from the client id obtained in the previous step that will be used as Audience Principal ID).

NOTE: For this step the Azure Portal user will need to have the necessary permissions to consent to grant permissions to the app. In the previous sample URL we are requesting read permissions for all SharePoint sites, this would require Manage permissions on the resources requested.
To make a request restricted to a specific site a URL like the following could be used requesting read permissions on an individual site:

```
https://<tenant_name>.sharepoint.com/sites/<sitename>/_layouts/15/OAuthAuthorize.aspx?
client_id=<client_id>&scope=Site.Read&response_type=code&redirect_uri=<redirect_uri>
```

Navigate to the URL from your browser. Login to the site if you have not logged in already. This opens a consent page that prompts the user to grant (or deny) the app the permissions that the app requests.



Once you grant the permission (by clicking trust), the SharePoint Online site asks the Access Control Service (ACS) to create a short-lived (approximately 5 minutes) authorization code unique to this combination of user and app. ACS sends the authorization code to the SharePoint site.

SharePoint Online site redirects the browser back to the redirect URI that was specified when the app was registered. It also includes the authorization code as a query string. The redirect URL is structured like the following:
`http://localhost:9090/oauth/2.0/redirectURL.jsp/?`**`code=<authcode>`** or
`https://<hostname>:9443/oauth/2.0/redirectURL.jsp/?`**`code=<authcode>`**

2.1.3 Get the access token and refresh token

Construct the below POST request:

- URL:
  - `https://accounts.accesscontrol.windows.net/<site_realm>/tokens/OAuth/2`
- Header:

- `Content-Type = "application/x-www-form-urlencoded"`

- Post parameters (in the body of the request):
  - `grant_type=authorization_code`
  - `client_id=<client_id>@<site_realm>`
  - `client_secret=<client_secret>`
  - `code=<auth_code>`
  - `redirect_uri=<redirect_uri>`
  - `resource=<audience_principal_ID>/<site_host>@<site_realm>`
  - Where:
    - `<site_realm>` is the Bearer realm obtained in the step "Get the Realm of the site".
    - `<client_id>` is <client id when registering the app>@<site realm from the step "Get the Realm of the site>.
    - `<client_secret>` is the client_secret obtained when registering the app.
    - `<auth_code>` is the auth code obtained in step "Get the authorization code".
    - `<resource>` is <audience principal ID>/<sharepoint domain>@<site realm>.
    - `<redirect_uri>` is the redirect URL set when registering the app.
    - `<audience_principal_ID>` is a permanent security principal ID for SharePoint. <audience_principal_ID> is obtained in step "Get the Realm of the site" (the value "`client_id`" in the response header `WWW-Authenticate`).

NOTE: all values need to be URL encoded (including the `client_secret`)

2.2 USING DENODO TEMPLATES FOR SHAREPOINT

If you are using the Denodo Templates for Sharepoint you can skip the following steps and return to the Denodo Templates for SharePoint - Quick User Guide.

2.3 USING THE ODATA 2 CUSTOM WRAPPER

First of all, you need to download the **Denodo OData 2 Custom Wrapper** from the Support Site and import it into the Virtual DataPort Server following the instructions provided in the documentation.

Once the OData Custom Wrapper is imported, create a new Custom Data Source selecting `New > Data Source > Custom` in the contextual menu.

- On "Select Jars" select the imported extension.

- Fill the "Class name" field with: `com.denodo.connect.odata.wrapper.ODataWrapper`
- Click the **Refresh Input Parameters** to show the parameters related to the wrapper to be configured.

The configuration of these parameters depends on the type of the authentication chosen.

2.3.1 Connecting through OData 2 using OAuth 2.0

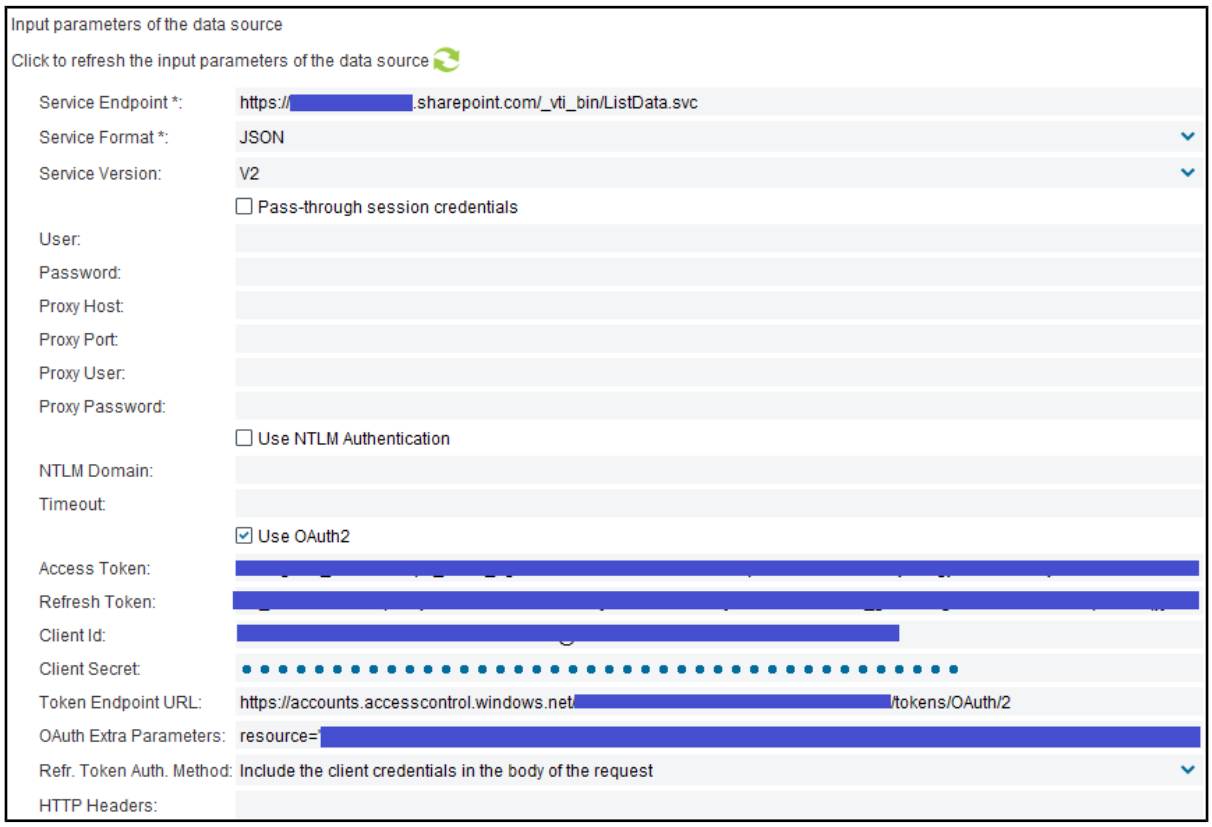
The following wrapper parameters need to be configured as follow to define datasource:

`Service Endpoint`	`https://<tenant_name>.sharepoint.com/_vti_bin/ListData.svc` Replace `<tenant_name>` with your actual tenant name
`Service Format`	`JSON or XML-ATOM`
`Service Version`	`V2`
`Use OAuth2`	`true`
`Access Token`	`access_token` obtained in the **Obtain OAuth tokens** section
`Refresh Token`	`refresh_token` obtained in the **Obtain OAuth tokens** section
`Client Id`	`<client_id>\@<tenant_id>` `<client_ id>` and `<tenant id>` from registering the app in **Configuration the application in Azure** section
`Client Secret`	`client secret` from registering the app in **Configuration the application in Azure** section
`Token Endpoint URL`	`https://accounts.accesscontrol.windows.net/<tenant_id>/tokens/OAuth/2` Replace `<tenant_id>` with the actual directory tenant id obtained in **Configuration the application in Azure** section
`OAuth Extra Parameters`	`resource="<audience_principal_id>/<tenant_name>.sharepoint.com\@<tenant_id>"` Replace `<tenant_id>` with the actual directory tenant id obtained in **Configuration the application in Azure** section and `<audience_principal_id>` with the value obtained in **Obtain OAuth tokens** section
`Refr. Token Auth. Method`	`Include the client credentials in the body of the request`

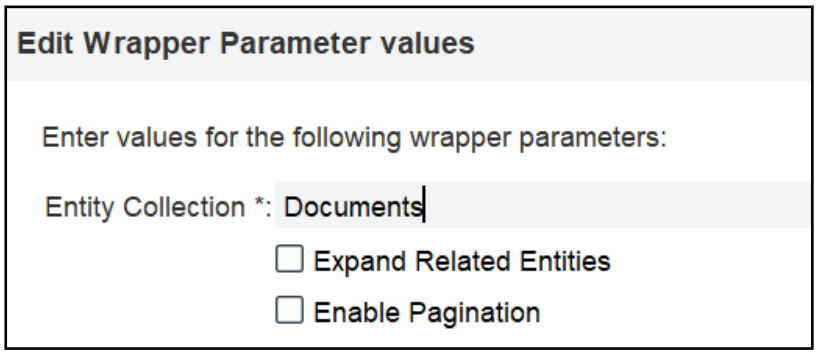
Fill in `Proxy` fields if required.

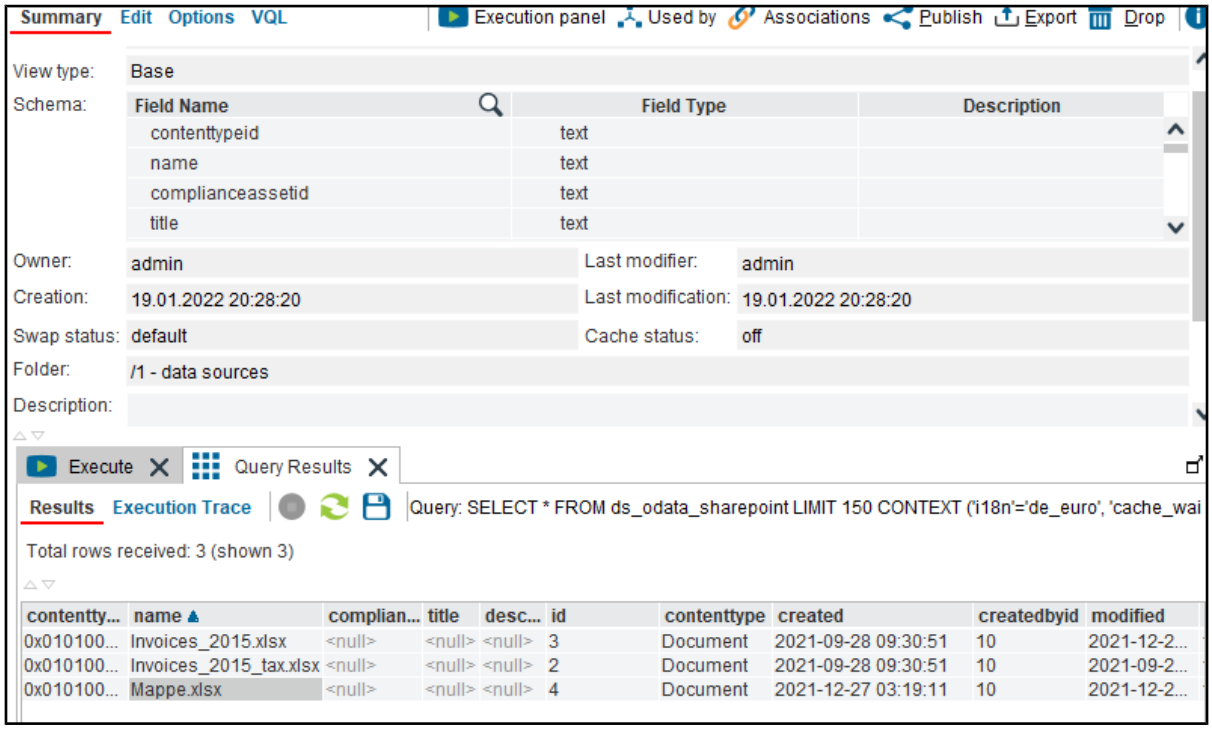
NOTE: the '@' must be escaped with '\' to prevent it from being confused with an interpolation variable.

This is how it will look like in the OData 2 Custom Wrapper:



Save the changes and select the Create base view option. This will display the Edit Wrapper Parameter Values dialog. For example you could use `Documents` as Entity Collection which will list you the elements you have in that site:





You can define any available collection name in the Entity Collection textbox .To know the list of available collections, browse the service endpoint URL used in the OData 2 Custom Wrapper datasource configuration

URL	`https://<tenant>.sharepoint.com/_vti_bin/ListData.svc/`

or provide a non-existent collection and the wrapper will list the available ones:

> There was an error while creating this base view: Error while executing custom wrapper method 'getSchemaParameters': Entity Collection not found for the requested service. Available Entity Collections are [SolutionGallery, ThemeGallery, Appfiles, Appdata, MasterPageGalleryCompatibleSearchDataTypes, MasterPageGallery, Attachments, SitePages, Events, ListTemplateGallery, Books, MasterPageGalleryTargetControlType, MasterPageGalleryTemplateLevel, SearchConfigList, WebPartGalleryGroup, ComposedLooks, MasterPageGalleryStandalone, WebPartGalleryRecommendationSettings, SharePointHomeOrgLinks, StyleLibrary, WebPartGallery, SampleLibrary, UserInformationList, MasterPageGalleryCompatibleUIVersionS, Documents, SearchConfigListScope, SitePagesSitePageFlags, TaxonomyHiddenList, FormTemplates, EventsCategory, MasterPageGalleryTargetControlTypeSearch]

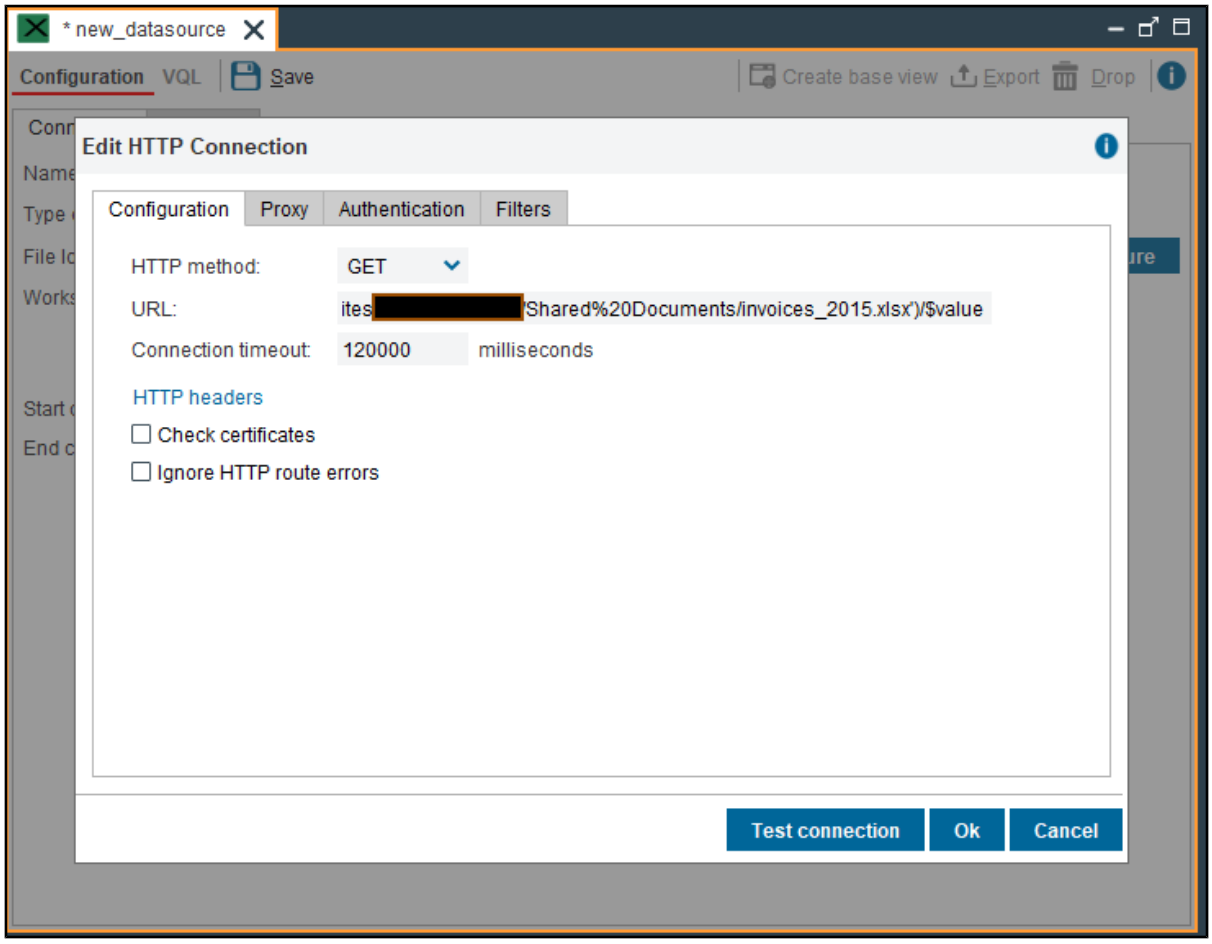
3 CONNECTING TO SHAREPOINT USING EXCEL, DELIMITED FILE, JSON AND XML DATA SOURCE

In this section we will explain how to create an Excel data source in Denodo to read Excel files from SharePoint. The same steps can be performed for JSON, XML or Delimited files as well.

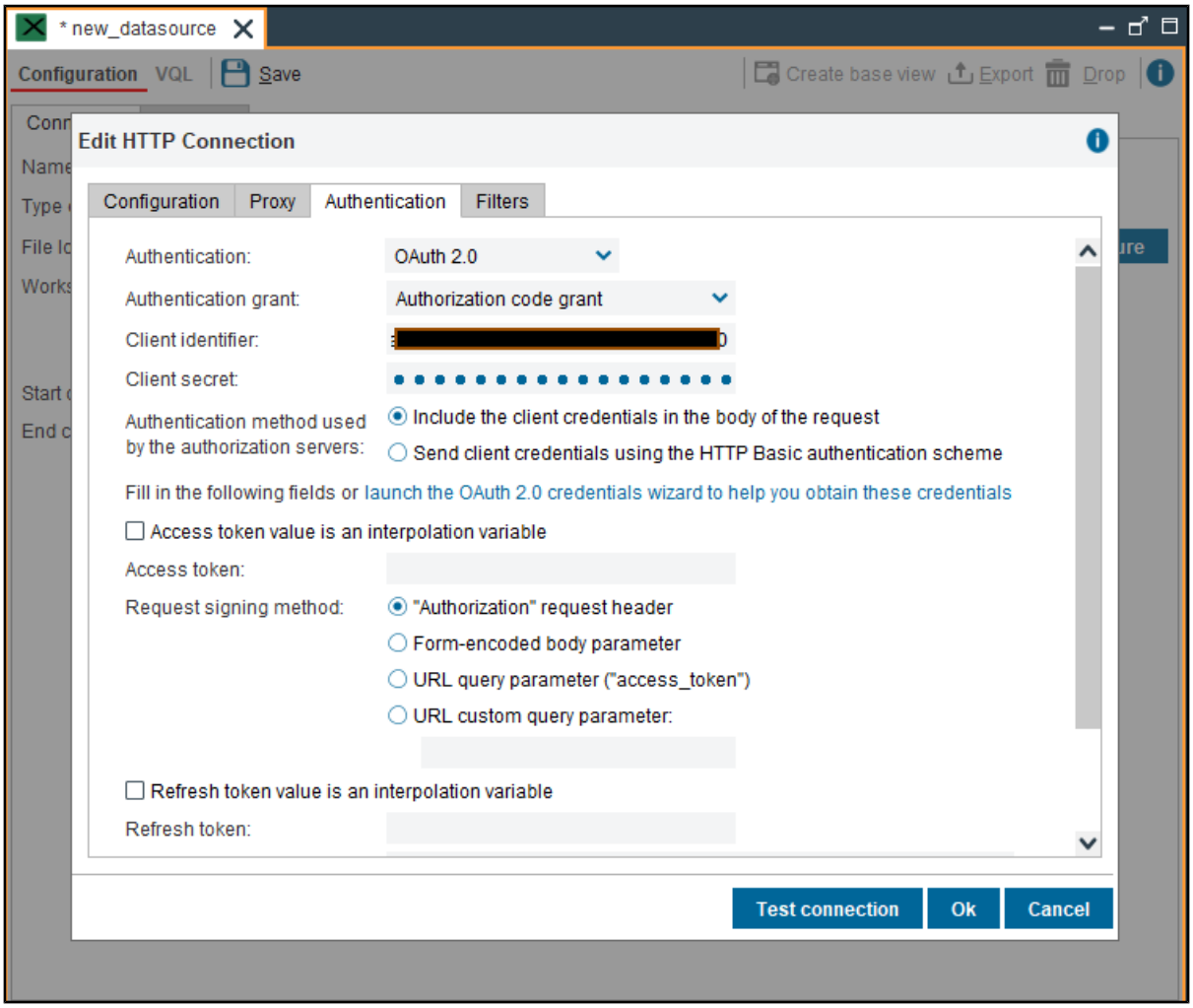
- From the Virtual DataPort Administration Tool or Design Studio, create an Excel Data Source.
- In the data source configuration, select the **`File Location`** as **`HTTP Client`** and select **`Configure`**.
- In the **`Edit HTTP Configuration`** dialog box set the `HTTP Method` to `GET` and provide the URL with the following format:

```
https://<tenant_name>.sharepoint.com/sites/site/_api/Web/GetFileByServerRelativePath(decodedurl='<location of an Excel file in the sharepoint>')/$value
```

Note: For some scenarios, the **site** parameter of the URL may need to be modified according to the company's Sharepoint structure. Review this URL with your Sharepoint administrator to make sure you use the correct one for your specific installation.



- In the Authentication section, select **`OAuth 2.0`** for **`Authentication`**.
- Paste the **`Application (client) ID`** from the newly created application to the **`Client Identifier`** and the **`Value of the Client Secret`** to the **`Client Secret.`**



- Then select the **`Launch the Oauth 2.0 credentials wizard`**. The Oauth 2.0 credentials wizard will pop up. See "Get the OAuth tokens using the OAuth credentials wizard" below.
- As for the **Token endpoint URL** use **`https://login.microsoftonline.com/<tenant_id>/oauth2/v2.0/token`** where **`<tenant_id>`** matches the tenant_id copied in one of the previous steps.
- Provide the Authorization server URL (again, replace **`<tenant_id>`** with the appropriate value).

```
https://login.microsoftonline.com/<tenant_id>/oauth2/v2.0/authorize
```

- Select the **`Redirect URI`**, provide the redirect URI that you used to register an application in the first step. Make sure that it matches.
- For the **Scopes** option provide the values

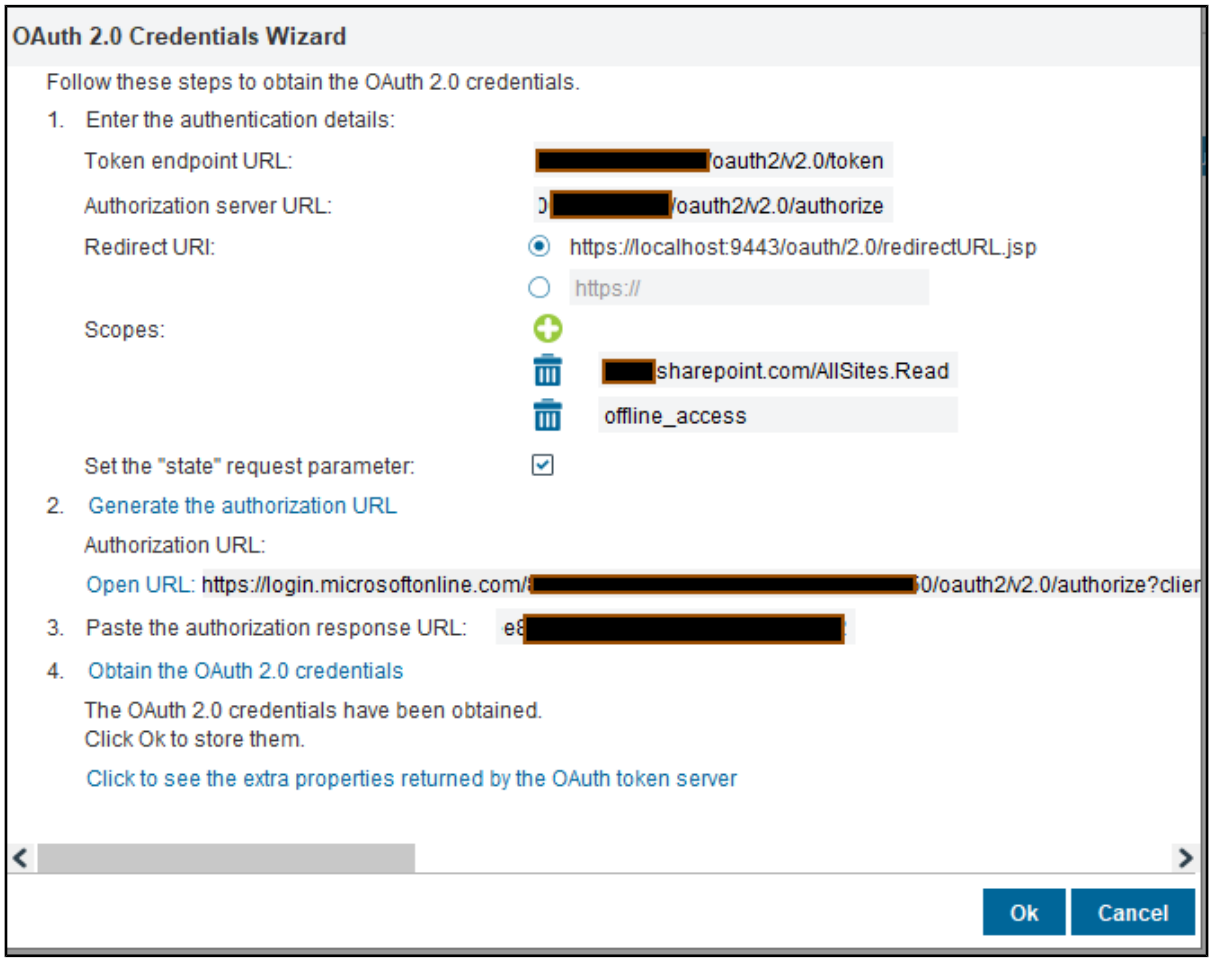
- **`https://<tenant_name>.sharepoint.com/AllSites.Read`**
- **`offline_access`**

Remember to replace **`<tenant_name>`** with your actual tenant name. The value **`offline_access`** is required in order to obtain the refresh token.

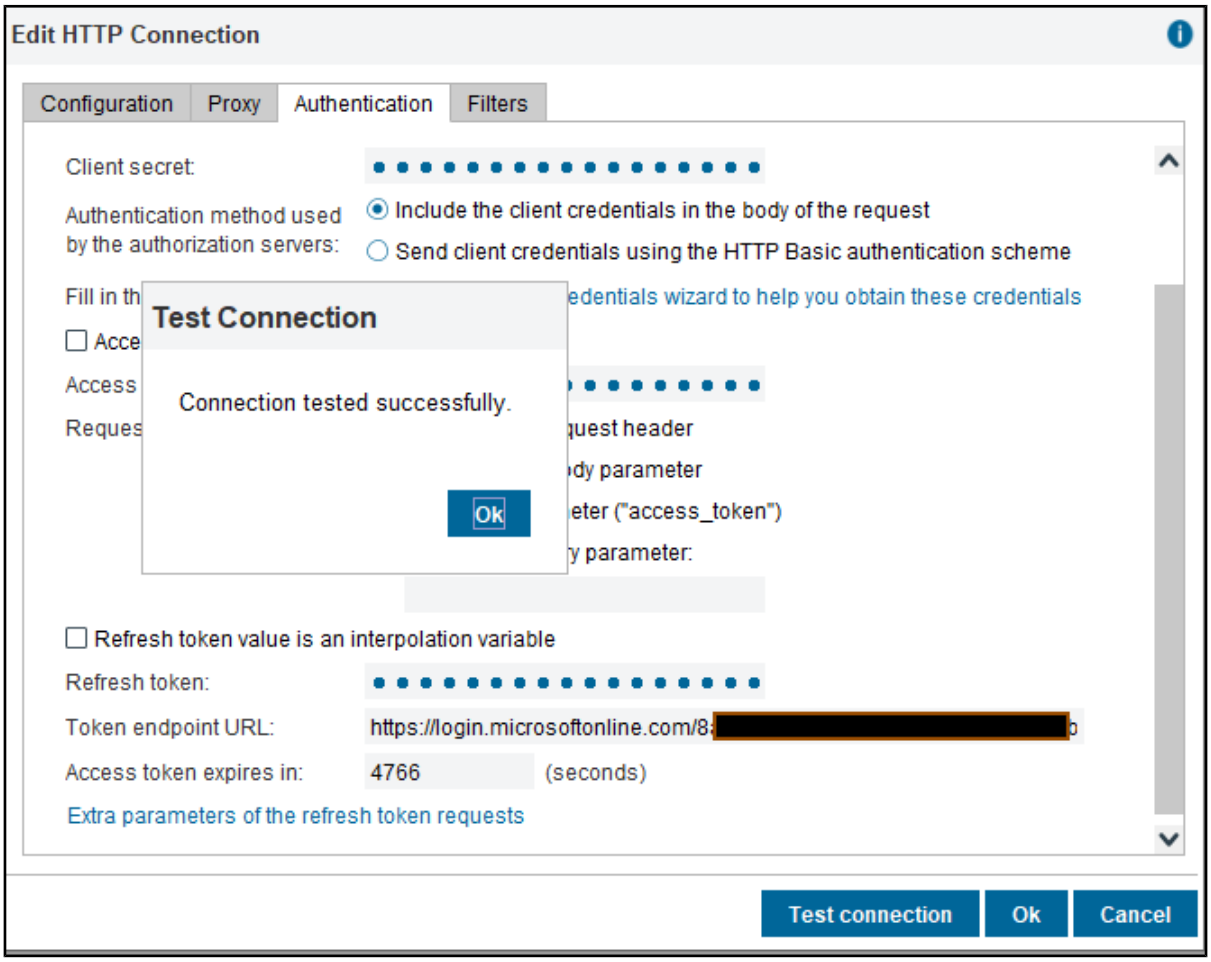
- After providing all the URL's click on the option **`Generate Authorization URL`**.
- When the Authorization URL is generated, click on the **Open URL**.
- Once the URL is opened on your desired browser, provide the permission, and once the permission is given the redirect URL with code would be generated. In this example, the format of the URL will be:

```
https://<hostname>:9443/oauth/2.0/redirectURL?
code=<>&state=<>&session_state=<>#
```

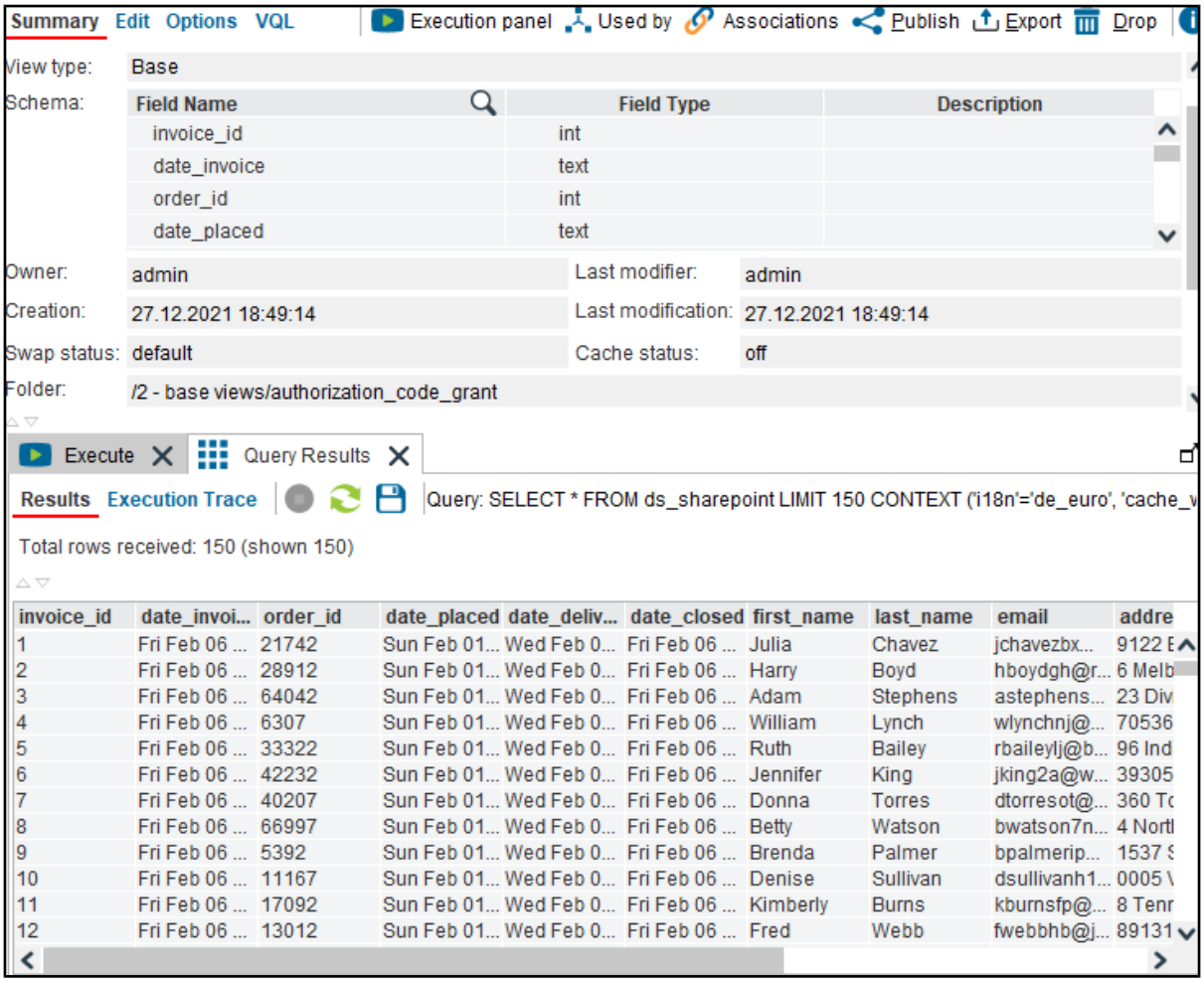
- Copy the full URL and paste the same on the **`Paste the authorized response URL`** and select **`Obtain the OAuth 2.0 credentials`**.
- Select **`Ok.`**



Once the OAuth 2.0 Credentials are obtained , test the connection to check if the connection can be established.

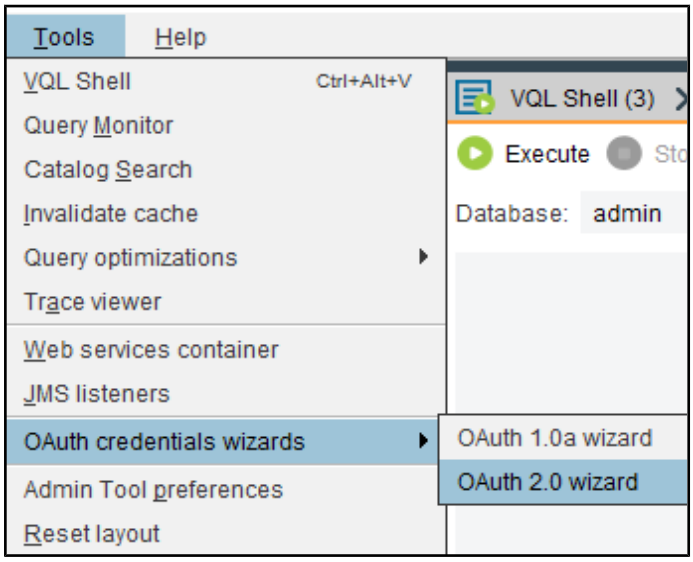


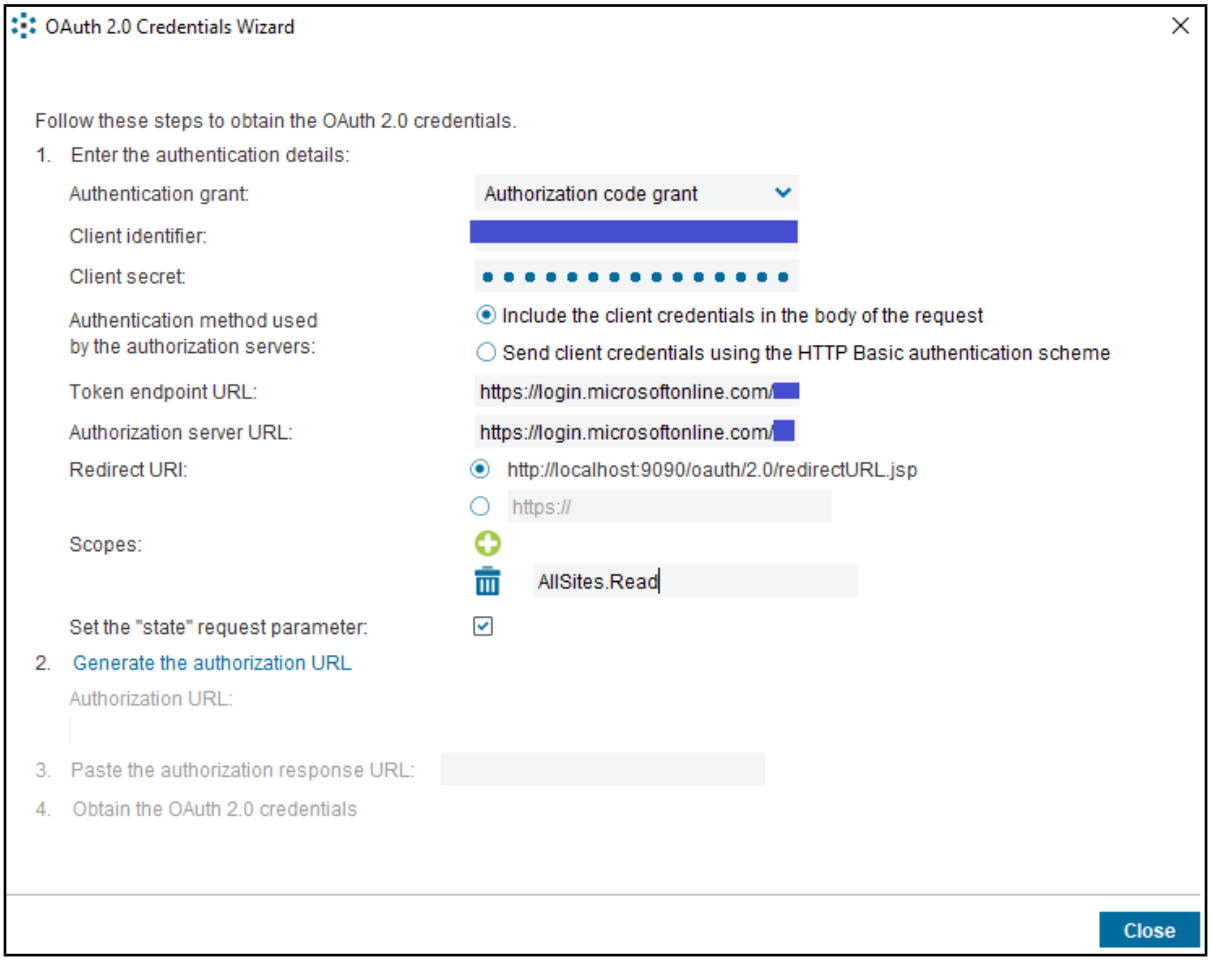
Now you will be able to create a new base view from this Excel data source.



3.1 GET THE OAUTH TOKENS USING THE OAUTH CREDENTIALS WIZARD

If you need to save the OAuth tokens to use for example in the JSON-based parts of the Denodo templates for SharePoint, you can generate them using the OAuth credentials wizard following the steps described below.





Where:

- Client identifier: the client id from the newly created Azure application
- Client secret: the value of the secret generated in the previous step for the newly created application.
- Token endpoint URL:

`https://login.microsoftonline.com/<tenant>/oauth2/token.`

- Authorization server URL : `https://login.microsoftonline.com/common/oauth2/authorize?resource=https:/<tenant_name>.sharepoint.com/.`
- Redirect URI: the default one.
- Scopes: add `AllSites.Read` and `offline_access.`

After providing all the values click on the option `Generate Authorization URL` and, when the Authorization URL is generated, click on the `open URL` link.

Once the URL is opened on your desired browser, provide the permission, and once the permission is given a URL with code would be generated. The format of the URL will be: `https://<tenant_name>.sharepoint.com/?code=<>&state=<>&session_state=<>#`

Paste the same on the `Paste the authorized response URL` and select `Obtain the OAuth 2.0 credentials`.

Finally, click on `Copy the credentials to the clipboard` to save the OAuth tokens.

4 APPENDIX I: AUTHENTICATING WITH NTLM

NOTE: This section only applies to SharePoint Server.

First of all, you need to download the **Denodo OData 2 Custom Wrapper** and import it as explained "Using the OData 2 Custom Wrapper" and then follow these steps.

To get the OData URL from the SharePoint Server the following steps are required:

- Open a browser and navigate to the following URL:

  ```
  http://<tenant>.sharepoint.com/_vti_bin/ListData.svc
  ```

  Replace `<tenant>` with your actual tenant name. This will be the **Service Endpoint.**

- Select the data feed you want to connect with. In the connection parameters this will be the **Entity Collection**.

- Finally it is necessary to determine the OData version (**Service Version**). You can do it by checking the following URL:

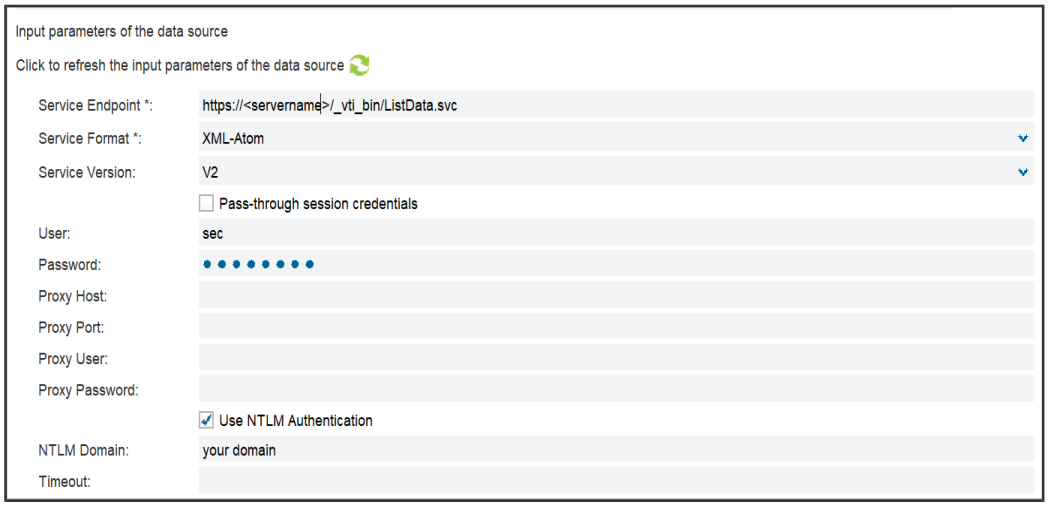
  ```
  https://<tenant>.sharepoint.com/_api/$metadata
  ```

The following wrapper parameters need to be configured as follow to define datasource:

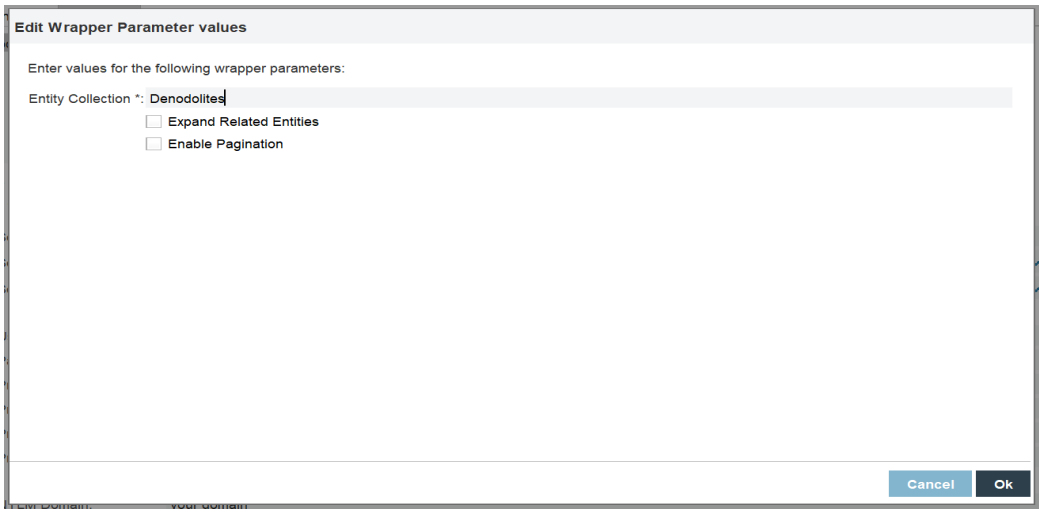
`Service Endpoint`	`https://<tenant_name>.sharepoint.com/_vti_bin/ListData.svc` Replace `<tenant_name>` with your actual tenant name
`Service Format`	`XML-Atom`
`Service Version`	Obtained in the previous step
`User`	Your `NTLM` user
`Password`	Your `NTLM` password
`Use the NTLM Authentication`	`true`
`NTLM Domain`	Your `NTLM` domain

Fill in `Proxy` fields if required.

This is how it will look like in the OData 2 Custom Wrapper:



Save the changes and select the `Create base view` option. This will display the `Edit Wrapper Parameter Values` dialog. Use the data feed selected as Entity Collection to query:



5 APPENDIX II: CONNECTING TO LISTS AS RSS FEEDS

NOTE: This section only applies to SharePoint Server.

To get the RSS URL from the SharePoint Server the following steps are required:

- Navigate to the site where the List is stored.
- On the left top corner of the screen select `Site actions - View All Site Content`.
- Select the list that is going to be used.
- Once the list has been selected open the `List Tools` menu and select the `List` option.
- In order to access the list using the RSS feed click on the RSS feed icon of the top menu and copy the RSS URL.

Once you have the RSS URL it can be imported into VDP using an XML data source:

- Create a new XML Data source selecting `New > Data Source > XML` in the contextual menu.
- Select "`Http Client`" as "`Data route`" and click on the "`Configure`" button.
- On the "`URL`" field insert the RSS URL.
- Fill the "`Authentication`" Tab if required and click the "`OK`" button. By default, SharePoint Server uses NTLM authentication but other authentication modes might be configured.
- Create the Base View.
- Execute the view and get the results.

Although the recommended ways to import data from Microsoft SharePoint are through the RSS interface and OData, there are other possible ways to import information:

- Using the List XML with its List GUID.
- Using the SharePoint Web Services (WSS).
- Creating an ITPilot wrapper to perform a web extraction on the SharePoint web interface.

These are more complex options and are not recommended unless the recommended options do not contain all the information that has to be imported into Virtual DataPort.

6 REFERENCES

Quickstart: Register an application with the Microsoft identity platform

Denodo OData2 Custom Wrapper - User Manual

Denodo Templates for SharePoint - Quick Use Guide