



# White Paper

## Fabasoft Integration for OData

2021 September Release

Copyright © Fabasoft R&D GmbH, Linz, Austria, 2021.

All rights reserved. All hardware and software names used are registered trade names and/or registered trademarks of the respective manufacturers.

No rights to our software or our professional services, or results of our professional services, or other protected rights can be based on the handing over and presentation of these documents.

## Contents

<b>1 Introduction</b>	<b>4</b>
<b>2 Installation of the Fabasoft OData Service</b>	<b>4</b>
<b>3 Basic Functionality</b>	<b>4</b>
3.1 Creating an OData Service .....	4
3.2 Creating a Password for Applications.....	5
<b>4 Access the OData Service via Microsoft Power BI</b>	<b>5</b>
<b>5 Access the OData Service as a Developer</b>	<b>8</b>
<b>6 Performance</b>	<b>9</b>
6.1 Performance of the Initial Call to an OData Service .....	9
6.2 Performance of Subsequent Calls to OData Service .....	9
<b>7 Restrictions</b>	<b>9</b>

## 1 Introduction

With the Fabasoft Integration for OData (<https://www.odata.org>), structured data can be accessed.

This document describes the setup of an OData service as well as the access via Microsoft Power BI. For detailed instructions on how to generate sophisticated reports with charts, lists, relations, etc., please check out the official Microsoft Power BI documentation.

## 2 Installation of the Fabasoft OData Service

For the time being, the Fabasoft OData Service functionality must be enabled by Fabasoft Cloud Sales ([cloud@fabasoft.com](mailto:cloud@fabasoft.com)).

## 3 Basic Functionality

An OData service can be created in apps that have reporting enabled (e.g. Scrum). The OData service contains the definition of the object classes and properties that are exposed via the OData API.

Depending on where you create the OData service, various source objects can be used. For example, in order to gain access to your contracts, create an OData service inside your Contract Manager app.

Having created an OData service, you receive a URL which can then be used to access the data via a client like Microsoft Power BI. The OData protocol is a standardized protocol which is understood by any client - even a web browser.

Aside from the OData URL, you require the "Access for Applications" password for authentication.

### 3.1 Creating an OData Service

Once reporting has been enabled in the app of your choice, you need to create the OData service in the context where your data is stored so that it can be queried.

To create an OData service, proceed as follows:

1. Navigate into the desired app configuration or app room, then into the "Reports and OData Services" area.
2. Click the "Create OData Service" action.
3. Define the following settings and click "Next".
  - Name  
The name of the OData service.
  - Language  
Multilingual strings are provided in the defined language in the data source.
  - Use Programming Name  
If enabled, object classes and properties will not be translated and the programming names are used instead.
  - Cache Duration  
Defines how long the data is cached. Defining longer time intervals will increase performance during subsequent calls.

4. Choose the object class of which you want to add properties to the model for the OData service and click "Next".
5. Select the desired properties and click "Next".  
**Note:** You need to consider that adding a great number of properties will require additional resources to load. It is recommended to add the minimum required number of classes and properties.
6. Click "Next".
7. If needed, you can add further properties of other object classes by clicking "Add Entry".
8. Click "Next".
9. Copy the provided link to the OData service and click "Next".  
**Note:** You will be able to copy this link later as well by clicking on the generated OData service.

## 3.2 Creating a Password for Applications

You need a password for applications to access your OData service. Create a password by opening the account menu (your user name) and clicking "Advanced Settings" > "Access for Applications". Create a password valid for "Open Data Protocol (OData)".

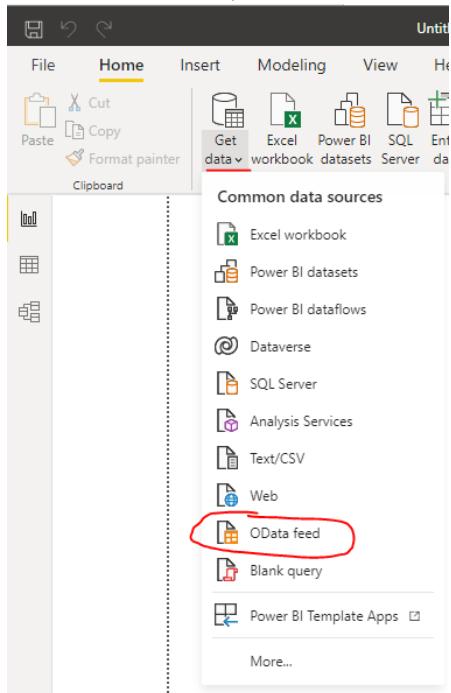
## 4 Access the OData Service via Microsoft Power BI

Using Microsoft Power BI to access the data provided by the OData service is quick and easy. Microsoft Power BI is a great tool for creating reports and statistics.

For more details on how to work with Microsoft Power BI please refer to:  
<https://powerbi.microsoft.com/en-us/support/>

To add the OData service to Microsoft Power BI, proceed as follows:

1. On the “Home” tab, click the “Get data” button, and then click “OData feed” as data source.



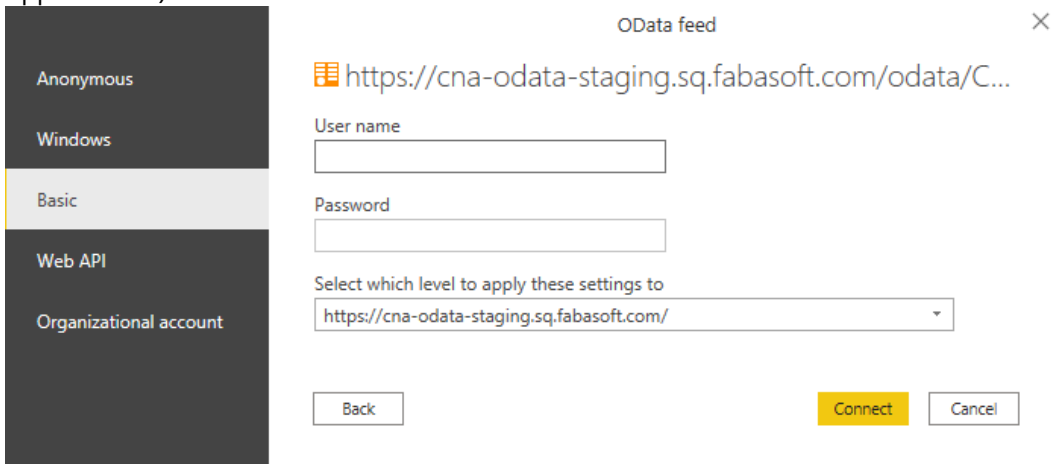
2. Enter the URL of the OData service (see chapter 3.1 “Creating an OData Service”) in the *URL* field and click “OK”.

**Note:** The “Basic” connection option is sufficient.

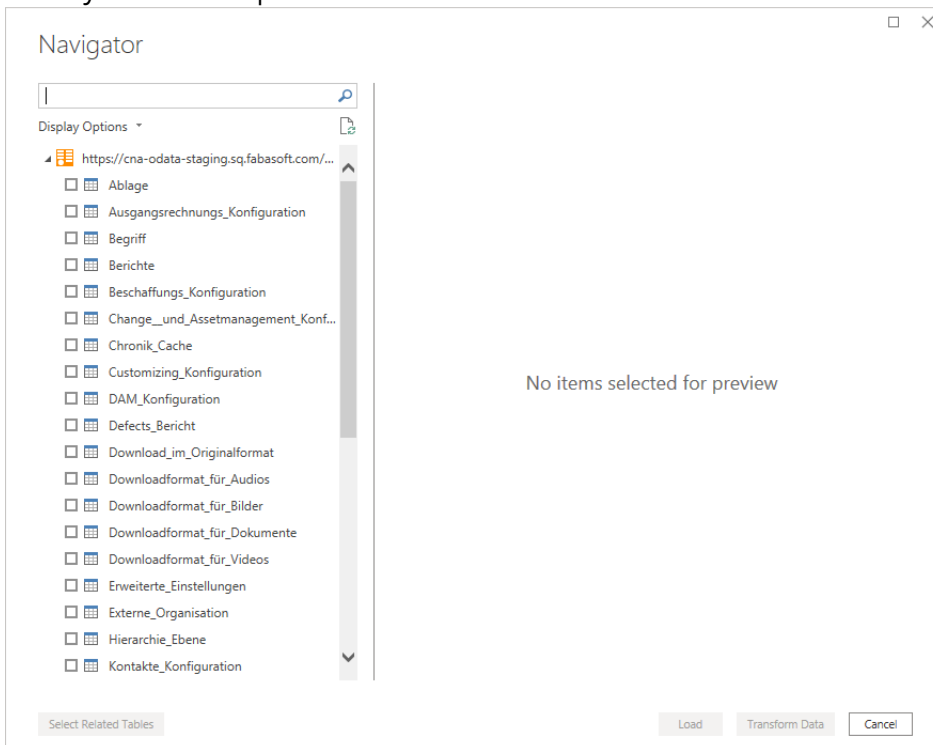


3. Since OData usage requires authentication you need to select the “Basic” authentication option. Enter your e-mail address for the cloud log-in as the user name and the generated Password for Applications as the password (see chapter 3.2 “Creating a Password for

Applications”). Click “Connect”.



4. After successful authentication, you can select from several display options in the “Navigator”. Select your desired options and click the “Load” button.

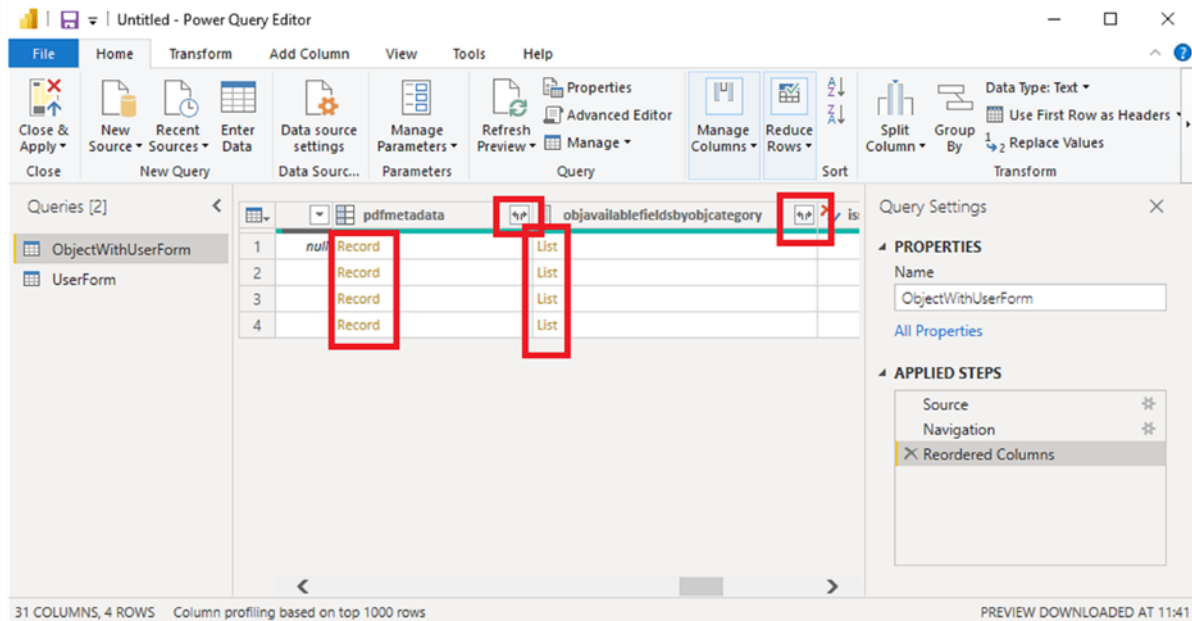


## Aggregates, Enumerations and Lists

When you create an OData service, it is very likely that some of the required properties have aggregates or lists as data types.

Aggregates are displayed as “Record” in Microsoft Power BI and lists are displayed as “List”. In the “Transform Data” dialog you can see these aggregates and lists.

In order to make them useable, click on the “Expand” tool at the end of the column (see red box on the column in the screenshot) to choose an expanding option. Such an option is e.g. creating a new table; however, this requires additional work as relationships between the tables need to be set.



The easiest way for aggregates (or “Records” in Microsoft Power BI terms) is to simply expand the columns: new columns are inserted for the chosen fields inside the aggregate. You can use it after clicking “Close & Apply”.

As for lists, the approach chosen depends on the expected number of entries. If you can predict having only a couple of entries, expanding the list is an option. In this case, the other values are duplicated. If there are many entries, creating a new table based on the list may be a better solution.

## 5 Access the OData Service as a Developer

The OData protocol is a standardized HTTP-based protocol to retrieve data. Please check <https://www.odata.org/> for details regarding the standard, programming examples and more.

When accessing an OData service URL via a web browser, you can interactively see the results of OData queries as JSON documents or XML documents.

Accessing data via an OData service is quite simple. Basically, all you need is to create an HTTP-based request with basic authentication turned on. The user name is the e-mail address and the password is the one created via “Password for Applications”.

The response is always a JSON document. In many programming languages, there is already a good toolset to parse and interpret JSON.

**Note:** If you have to process the metadata as well, simply follow the URLs stored in `@odata.context` which you can see in the screenshot below.



```

"@odata.context": "http://[REDACTED]/odata/COO.1.586.2.3542/$metadata#ObjectWithUserForm"
"value": [
  {
    "objaddress": "COO.1.586.3.3542",
    "objclass": "COO.1.1881.1.364527",
    "objname": "sadf",
    "objcreatedby": "COO.1.586.1.5158",
    "objcreatedat": "2021-07-21T18:53:48+02:00",
    "objchangedby": "COO.1.586.1.5158",
    "objmodifiedat": "2021-07-21T18:53:48+02:00",
    "objsecchangedat": "2021-07-21T18:53:48+02:00",
    "objchangedat": "2021-07-21T18:53:48+02:00",
    "objuserchangedat": "2021-07-21T18:53:48+02:00",
    "objowner": "COO.1.586.1.5899",
    "objowngroup": "COO.1.586.1.5897",
    "objaobj": "COO.1.1881.1.331843",
    "objectversnr": 1,
    "objectverscreated": "2021-07-21T18:53:48+02:00",
  }
]

```

## 6 Performance

The Fabasoft Integration for OData processes data in of two steps:

- During the first step (“Cube creation”), all objects as defined in the data source are queried and loaded into the OData service. The scope of objects considered is defined by the context of the data source (e.g. Teamroom). If the objects of the query result contain references to instances of classes contained in the data source, these instances are added to the object set. No queries are executed for the object classes “Object”, “User” and “Group”. Instead, the references to these instances are collected in the corresponding tables.
- During the second step (“Table creation”), the values per table are loaded and mapped to the structure as required by the OData query processor. This data is created per table on demand. However, all of the table rows are created independent of the amount of table rows queried.

Keep in mind that any caching within the OData services is performed per user. Thus, if multiple users access an OData service, they all work with their dedicated OData cache.

### 6.1 Performance of the Initial Call to an OData Service

In order to reduce the processing overhead of the initial call to an OData service you can

- reduce the number of entities or tables, and/or
- reduce the number of properties inside entities or tables with high row count.  
Try removing unused properties to reduce the overall processing time for each entity. This is most likely the fastest way to resolve such performance issues.

### 6.2 Performance of Subsequent Calls to OData Service

Should the OData service lack performance with regards to subsequent calls, you may increase the value in the *Cache Duration* field. In this case, the data is stored in the cache for a longer period, hence subsequent calls may reuse the already existing data table.

## 7 Restrictions

There is no support for content or dictionary properties.