



PUBLIC

Document Version: 2026.18 – 2026-08-25

Consuming Data Exposed by SAP Datasphere

Content

1	Consuming Data Exposed by SAP Datasphere.	3
1.1	Consume Data in SAP Analytics Cloud via a Live Connection.	7
1.2	Integrate with SAP Analytics Cloud for Planning.	9
	Store and Consume SAP Analytics Cloud Data in SAP Datasphere.	10
	Consume SAP Datasphere Data in SAP Analytics Cloud via an OData Service.	11
	Load SAP Analytics Cloud Planning Data into SAP Datasphere.	13
1.3	Consume Data in Microsoft Excel via an SAP Add-In.	15
1.4	Consume Data in Power BI and Other Clients, Tools, and Apps via an OData Service.	17
1.5	Consume Data in Power BI and Other Clients, Tools, and Apps via ODBC/JDBC.	19
1.6	Consuming Data via the OData API.	20

1 Consuming Data Exposed by SAP Datasphere

All users with any of the standard roles can consume data exposed by spaces they are assigned to. If a user does not need to access SAP Datasphere itself, and only wants to consume data exposed by it, they should be granted a consumer role.

This topic contains the following sections:

- [Expose Data from SAP Datasphere \[page 3\]](#)
- [Consume Data in SAP Analytics Cloud \[page 4\]](#)
- [Integrate with SAP Analytics Cloud for Planning \[page 5\]](#)
- [Consume Data in Microsoft Excel \[page 5\]](#)
- [Consume Data via ODBC/JDBC \[page 6\]](#)
- [Consume Data via an OData Service \[page 6\]](#)

Expose Data from SAP Datasphere

Data can be exposed as analytic models, perspectives, and views, which are accessible to clients, tools, and apps as follows:

Object	SAP Analytics Cloud	Microsoft Excel	Other Clients, Tools, and Apps
Analytic models (see Creating an Analytic Model) Exposed: Automatically	Live Connection	Live Connection (via an SAP Add-In)	OData
Perspectives (see Define Perspectives) Exposed: Automatically	Live Connection	Live Connection (via an SAP Add-In)	-
Views* (see Exposing Data For Consumption) Exposed: When the <i>Expose for Consumption</i> switch is enabled	OData**	-	OData ODBC/JDBC

Object	SAP Analytics Cloud	Microsoft Excel	Other Clients, Tools, and Apps
For more information, see:	• Consume Data in SAP Analytics Cloud via a Live Connection [page 7] • Integrate with SAP Analytics Cloud for Planning [page 9]	• Consume Data in Microsoft Excel via an SAP Add-In [page 15]	• Consume Data in Power BI and Other Clients, Tools, and Apps via an OData Service [page 17] • Consume Data in Power BI and Other Clients, Tools, and Apps via ODBC/JDBC [page 19]

* The workflow of consuming views with a semantic usage of *Analytical Dataset* in SAP Analytics Cloud and Microsoft Excel via live connection is now deprecated. We recommend that you migrate your analytical datasets to the new *Fact* semantic usage and expose your view data via analytic models (see [Analytical Datasets \(Deprecated\)](#)).

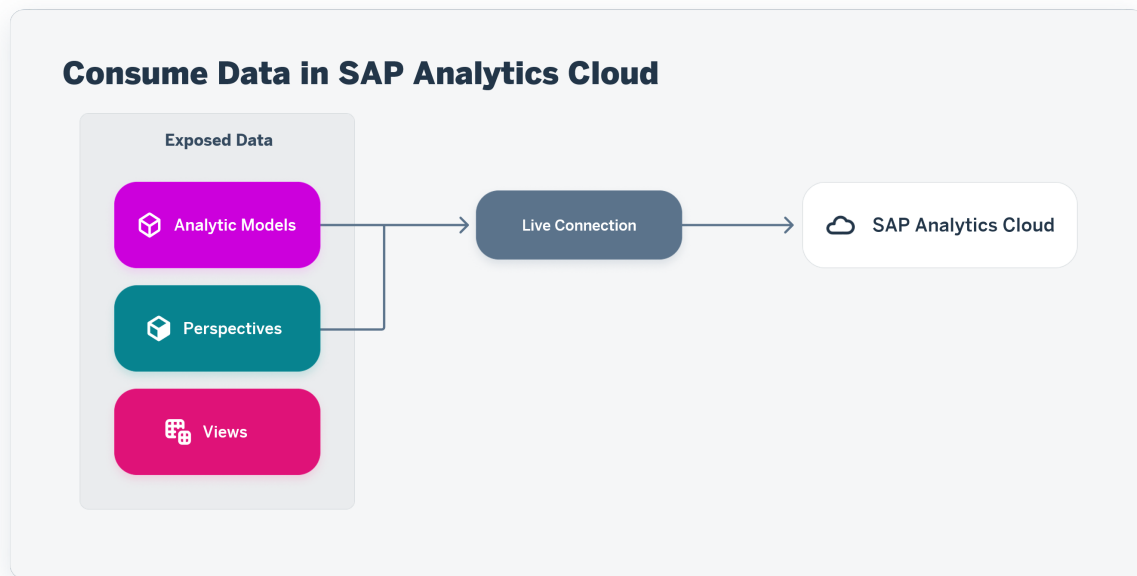
** SAP Analytics Cloud primarily uses the consumption of view data via OData for planning (see [Integrate with SAP Analytics Cloud for Planning \[page 9\]](#)).

Note

Before exposing data for consumption, you should consider applying row-level security via data access controls (see [Securing Data with Data Access Controls](#)).

Consume Data in SAP Analytics Cloud

You can create a live connection from SAP Analytics Cloud to SAP Datasphere and consume data exposed as analytic models and perspectives to create stories and analytic applications.



For more information, see [Consume Data in SAP Analytics Cloud via a Live Connection \[page 7\]](#).

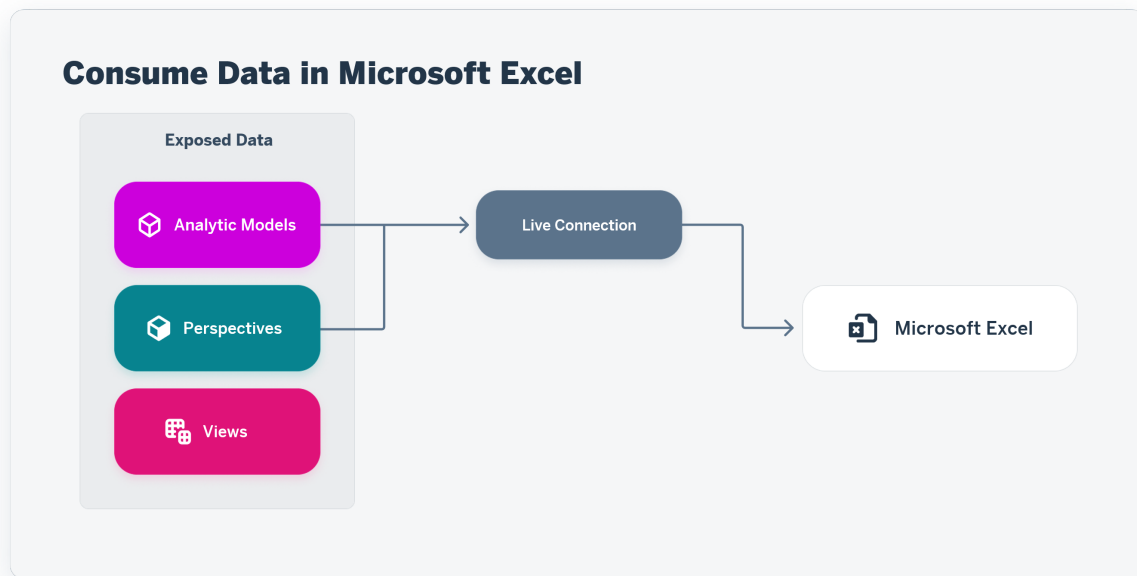
Integrate with SAP Analytics Cloud for Planning

You can use SAP Datasphere as a data source for loading actuals or external data into an SAP Analytics Cloud planning model. You can also load your SAP Analytics Cloud planning data into SAP Datasphere and combine it with live actuals or other data as appropriate.

For more information, see [Integrate with SAP Analytics Cloud for Planning \[page 9\]](#).

Consume Data in Microsoft Excel

You can create a live connection from SAP Analytics Cloud to SAP Datasphere and consume data exposed as analytic models and perspectives in Microsoft Excel, via the SAP Analytics Cloud, add-in for Microsoft Excel.



For more information, see [Consume Data in Microsoft Excel via an SAP Add-In \[page 15\]](#).

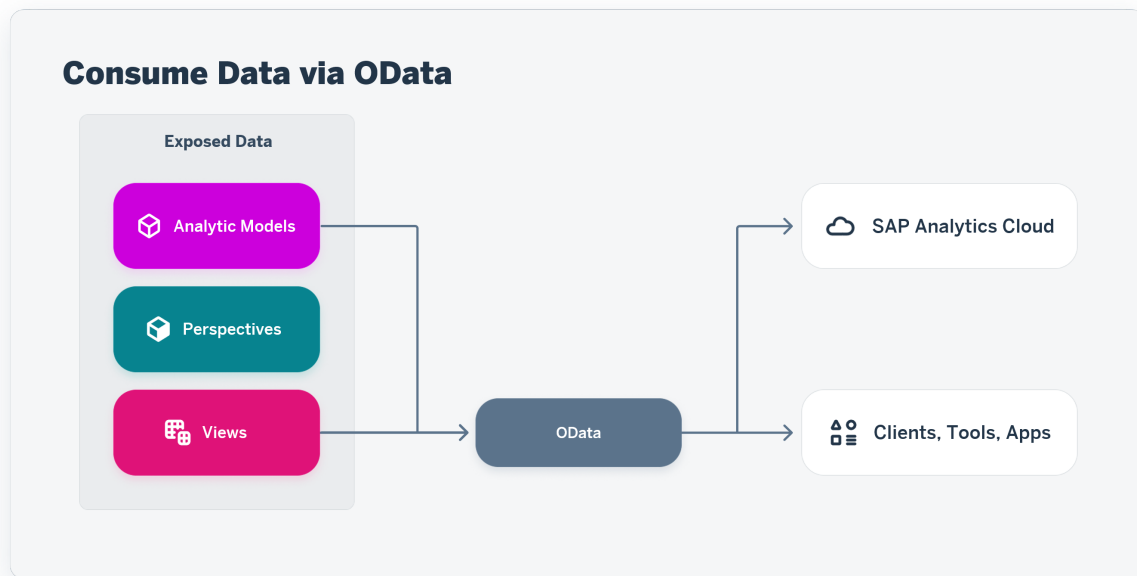
Consume Data via ODBC/JDBC

You can consume data exposed from SAP Datasphere in clients, tools, and apps via a database user/Open SQL schema.

See [Consume Data in Power BI and Other Clients, Tools, and Apps via ODBC/JDBC \[page 19\]](#).

Consume Data via an OData Service

You can connect to the OData API and consume data exposed as views or analytic models in SAP Analytics Cloud and other clients, tools, and apps that are capable of accessing an OData service and authenticating via an OAuth client.



For more information, see [Consume Data in Power BI and Other Clients, Tools, and Apps via an OData Service \[page 17\]](#) and [Consuming Data via the OData API \[page 20\]](#).

1.1 Consume Data in SAP Analytics Cloud via a Live Connection

You can create a live connection from SAP Analytics Cloud to SAP Datasphere and consume data exposed as analytic models and perspectives to create stories and analytic applications.

Prerequisites

To consume data exposed by SAP Datasphere in SAP Analytics Cloud, you must have a scoped role that grants you access to a space with the following privileges:

- [Data Warehouse Consumption](#) (-R- - - - -) - To access data exposed by SAP Datasphere spaces.
- [Space Files](#) (-R- - - - -) - To access objects in spaces.

The [DW Consumer](#) role template grants these privileges, and is intended for users who will only consume data and do not need to connect to SAP Datasphere itself. All other standard SAP Datasphere role templates also grant these privileges. For more information, see [Privileges and Permissions](#) and [Standard Roles Delivered with SAP Datasphere](#).

📌 Note

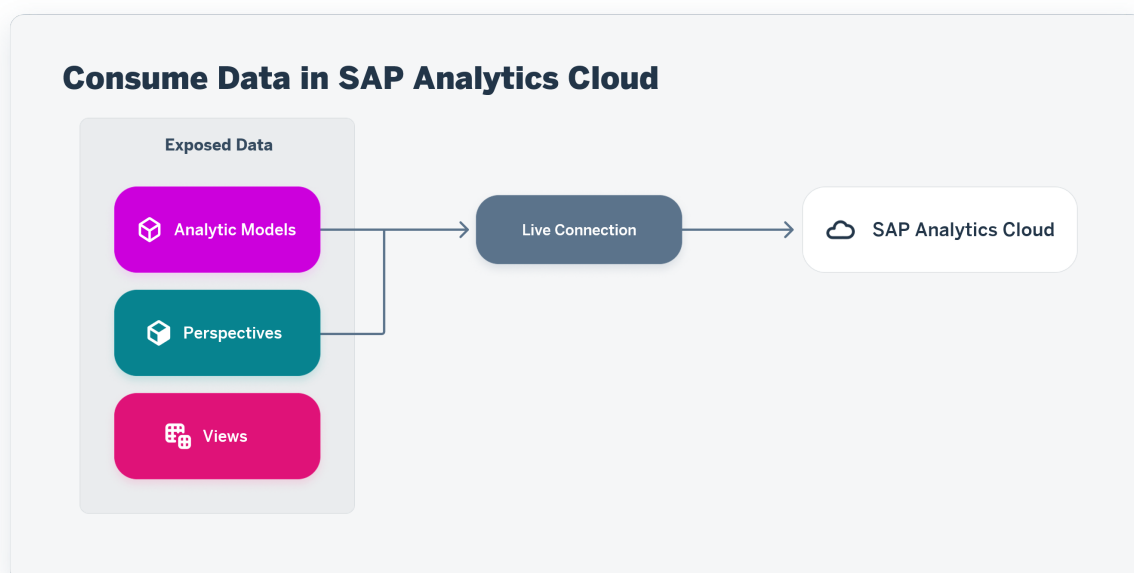
If data access controls are applied, then the data you can consume is filtered based on your user id (see [Securing Data with Data Access Controls](#)).

You must, in addition:

- Be an SAP Analytics Cloud user with the role *BI Content Creator* or another role providing equivalent privileges.
- Create or have access to an SAP Analytics Cloud live data connection to your SAP Datasphere tenant (see [Live Data Connections to SAP Datasphere](#) in the *SAP Analytics Cloud* documentation).

Note

The workflow of consuming views with a semantic usage of *Analytical Dataset* in SAP Analytics Cloud and Microsoft Excel via live connection is now deprecated. We recommend that you migrate your analytical datasets to the new *Fact* semantic usage and expose your view data via analytic models (see [Analytical Datasets \(Deprecated\)](#)).



Procedure

1. Navigate to your SAP Analytics Cloud tenant.

Note

If your SAP Datasphere is linked to an SAP Analytics Cloud tenant (and you are assigned one or more BI roles), you can use the  (*Product Switch*) in the top-right corner of the screen to navigate there.

2. In the side navigation area, select  (*Stories*).
3. Click the *Responsive* or *Canvas* tile, select the *Optimized Design Experience* mode, and click *Create* to open a new story in the editor.

Note

SAP Datasphere analytic models are not supported in the *Classic Design Experience* mode.

4. In the *Tools* menu, click *Add New Data* and then select *Data from an existing dataset or model* to open the *Select Dataset or Model* dialog.
5. In the panel on the left, select your SAP Datasphere live data connection, select your space, and then click the object to open.

Note

The workflow of consuming views with a semantic usage of *Analytical Dataset* in SAP Analytics Cloud and Microsoft Excel via live connection is now deprecated. We recommend that you migrate your analytical datasets to the new *Fact* semantic usage and expose your view data via analytic models (see *Analytical Datasets (Deprecated)*).

6. Drag a chart or table from the *Assets* list in the left panel and build your story as usual.

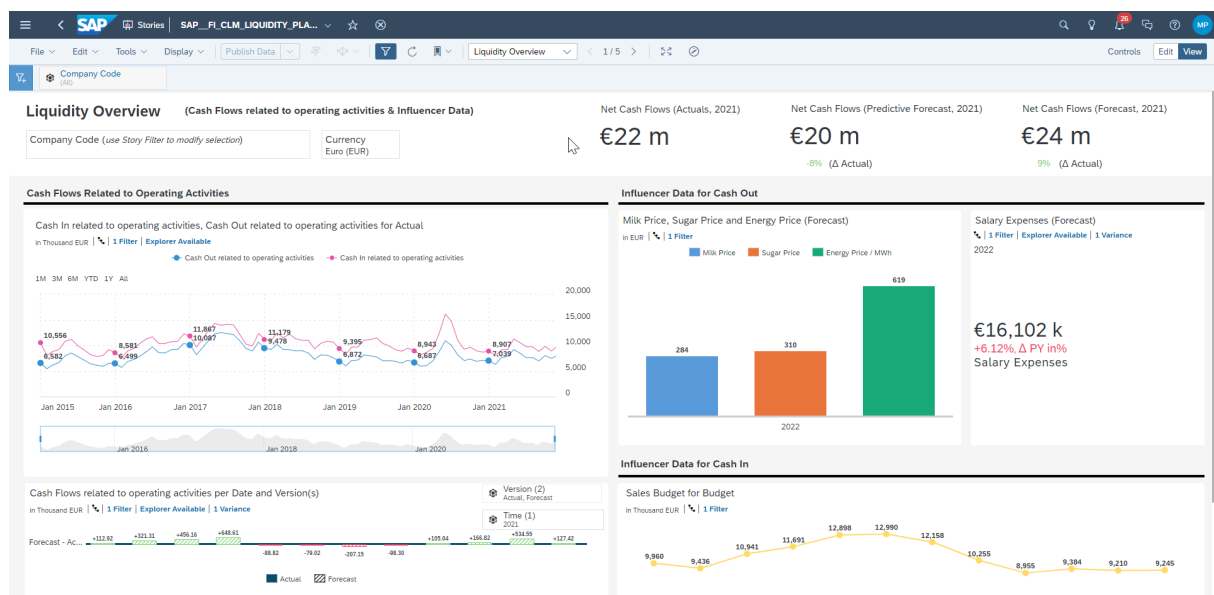
For detailed information about creating stories, see *Data Visualization (Stories)* in the *SAP Analytics Cloud* documentation.

Note

For information about SAP Analytics Cloud limitations with this type of connection, see SAP Note [2832606](#).

1.2 Integrate with SAP Analytics Cloud for Planning

SAP Datasphere integrates with SAP Analytics Cloud to act as a data source for loading actuals or external data into a planning model, and can also persist your planning data and combine it with live actuals or other data as appropriate.



You can:

- Load actuals data (or external data from the marketplace) from SAP Datasphere into SAP Analytics Cloud to use it as input for predictions or driver-based planning in your planning models via an OData Services Connection (see [Consume SAP Datasphere Data in SAP Analytics Cloud via an OData Service \[page 11\]](#)).
- Store data from SAP Analytics Cloud in SAP Datasphere and optionally combine it with actuals data or other sources as appropriate (see [Store and Consume SAP Analytics Cloud Data in SAP Datasphere \[page 10\]](#)).
- Reuse TCUR* views generated in SAP Datasphere for currency conversion in SAP Analytics Cloud planning models (see [Reuse SAP Datasphere Currency Rate Views in Seamless Planning Models](#) in the *SAP Analytics Cloud* documentation).

See also the blog [Seamless planning between SAP Analytics Cloud and SAP Datasphere](#)  (updated Dec 2024).

1.2.1 Store and Consume SAP Analytics Cloud Data in SAP Datasphere

SAP Analytics Cloud users can save their planning and analytic model data directly in SAP Datasphere spaces. The SAP Analytics Cloud users can choose for each object, whether to expose it to SAP Datasphere modelers for use as sources for views, flows, and analytic models.

This topic contains the following sections:

- [Expose SAP Analytics Cloud Objects in SAP Datasphere Spaces \[page 10\]](#)
- [Consume SAP Analytics Cloud Objects in SAP Datasphere Views and Flows \[page 11\]](#)

See also the blog [Seamless planning between SAP Analytics Cloud and SAP Datasphere](#)  (updated Dec 2024).

Expose SAP Analytics Cloud Objects in SAP Datasphere Spaces

SAP Analytics Cloud users can choose to save their model, public dimension, and currency rate table data in SAP Datasphere spaces.

For model and other data saved in an SAP Datasphere space, the SAP Analytics Cloud user can choose, for each object, whether to expose it for consumption in SAP Datasphere. Objects saved and exposed in this way can be consumed directly in views, flows, and analytic models without need for any replication between the two services.

For more information, see [Store SAP Analytics Cloud Planning Data in SAP Datasphere](#) and [Expose Objects for Consumption in SAP Datasphere](#) in the *SAP Analytics Cloud* documentation.

Consume SAP Analytics Cloud Objects in SAP Datasphere Views and Flows

When SAP Analytics Cloud objects saved in SAP Datasphere spaces are exposed, they appear as read-only tables:

- Models have a semantic usage of *Fact*.
- Public dimensions have a semantic usage of *Dimension*.

These local tables can be added as sources to views and flows (see [Add Objects from the Repository](#)).

Note

Replication flows do not support using objects exposed from SAP Analytics Cloud as sources.

For example, you could combine planning data saved from SAP Analytics Cloud with your actuals data (or any other appropriate data) in a view and then consume the view in an analytic model to be consumed in via a live connection (see [Consume Data in SAP Analytics Cloud via a Live Connection \[page 7\]](#)).

1.2.2 Consume SAP Datasphere Data in SAP Analytics Cloud via an OData Service

You can create an import data connection in SAP Analytics Cloud to consume data from an asset exposed via the SAP Datasphere OData API and consume it in an SAP Analytics Cloud model.

This topic contains the following sections:

- [Create an SAP Analytics Cloud OData Services Connection to an SAP Datasphere View \[page 11\]](#)
- [Create a Model from Your Connection \[page 13\]](#)

Create an SAP Analytics Cloud OData Services Connection to an SAP Datasphere View

You can create an OData services connection in SAP Analytics Cloud to load (replicate) data from an SAP Datasphere object to SAP Analytics Cloud.

Note

If the object being consumed is a view, it must be persisted and any remote tables in its lineage must have their data replicated to SAP Datasphere.

To load data from SAP Datasphere to SAP Analytics Cloud, via an OData service, you must have a scoped role that grants you access to a space with the following privileges:

- *Data Warehouse Consumption* (-R- - - - -) - To access data exposed by SAP Datasphere spaces.
- *Space Files* (-R- - - - -) - To access objects in spaces.

The *DW Consumer* role template grants these privileges, and is intended for users who will only consume data and do not need to connect to SAP Datasphere itself. All other standard SAP Datasphere role templates also

grant these privileges. For more information, see [Privileges and Permissions](#) and [Standard Roles Delivered with SAP Datasphere](#).

Note

If data access controls are applied, then the data you can consume is filtered based on your user id (see [Securing Data with Data Access Controls](#)).

You must, in addition:

- Obtain the following parameters for an OAuth client defined in your SAP Datasphere tenant with *Purpose* set to *Interactive Usage* or *Technical User*:

Standard OAuth2 Authorization Flow	OAuth2SAMLBearer Principal Propagation Flow
• <i>Client ID</i> • <i>Secret</i> • <i>Authorization URL</i> (not required for clients with a <i>Technical User</i> purpose) • <i>Token URL</i> Users of a client with a <i>Technical User</i> purpose (which includes its own privileges and permissions) can then access their resource directly. Users of clients with other purposes must manually authenticate against the IDP in order to generate the authorization code before continuing with the remaining OAuth2.0 steps.	• <i>Client ID</i> • <i>Secret</i> • <i>OAuth2SAML Token URL</i> • <i>OAuth2SAML Audience</i> Users authenticate with their third-party app, which has a trusted relationship with the IDP, and do not need to re-authenticate (see Add a Trusted Identity Provider).

For information about OAuth clients, see [Create OAuth2.0 Clients to Authenticate Against SAP Datasphere](#).

- Be an SAP Analytics Cloud user with the role *BI Content Creator* or another role providing equivalent privileges.
1. Navigate to your SAP Analytics Cloud tenant.

Note

If your SAP Datasphere is linked to an SAP Analytics Cloud tenant (and you are assigned one or more BI roles), you can use the  ([Product Switch](#)) in the top-right corner of the screen to navigate there.

2. In the side navigation area, select .
3. Click [Add Connection](#), select *OData Services* to open the *New OData Services Connection* dialog, and complete the fields as follows:

Parameter	Value
Connection Name	Enter an appropriate name for your connection.

Parameter	Value
Data Service URL	Enter service URL to your SAP Datasphere object in the following format: https://<tenant_url>.cloud.sap/dwaas-core/odata/v4/consumption/relational/<Space_ID>/<view> Note You can obtain the service URL from the Service tab of the Generate OData Request dialog in the view editor (see Consuming Data via the OData API [page 20]).
Authentication Type	Select OAuth 2.0 Authorization Code .
OAuth Client ID	Enter the values for your SAP Datasphere OAuth client (see Create an OAuth2.0 Client with an Interactive Usage Purpose).
Secret	
Token URL	
Authorization URL	

- Click [Create](#) to create the connection.

Create a Model from Your Connection

You can use your OData Services connection to load your object's data into an SAP Analytics Cloud model:

- In the side navigation area, click [Modeler](#).
- Click the [From a Data Source](#) tile and then select [OData Services](#).
- In the [Create Model from OData Services](#) dialog, select your connection in the list and click [Next](#). Follow the usual steps in the [New Query for OData Services](#) dialog (see [Building a Query](#) in the *SAP Analytics Cloud* documentation).

Note

Alternatively, you can load data into an existing SAP Analytics Cloud model (see [Import and Prepare Fact Data for a Model with Measures](#) in the *SAP Analytics Cloud* documentation).

1.2.3 Load SAP Analytics Cloud Planning Data into SAP Datasphere

You can transfer your SAP Analytics Cloud planning model data into SAP Datasphere via a Cloud Data Integration connection and optionally combine and compare it with your actuals data.

This topic contains the following sections:

- [Create an SAP Datasphere Cloud Data Integration Connection to Your SAP Analytics Cloud Tenant \[page 14\]](#)

- [Load SAP Analytics Cloud Planning Data to SAP Datasphere \[page 14\]](#)
- [Combine Actuals and Planning Data in an SAP Datasphere View \[page 15\]](#)

Note

Alternatively, for information about storing SAP Analytics Cloud data directly in SAP Datasphere without data movement, see [Store and Consume SAP Analytics Cloud Data in SAP Datasphere \[page 10\]](#).

Create an SAP Datasphere Cloud Data Integration Connection to Your SAP Analytics Cloud Tenant

You can create a Cloud Data Integration connection in SAP Datasphere to load (replicate) data from an SAP Analytics Cloud planning model to SAP Datasphere.

To load data from SAP Analytics Cloud to SAP Datasphere, you must:

- Be an SAP Analytics Cloud user with the role *BI Content Creator* or another role providing equivalent privileges.
- Obtain the following parameters for an OAuth client defined in your SAP Analytics Cloud tenant:
 - Client ID
 - Secret
 - Token URL
 - Authorization URL
- [if remote tables are required] Ensure that Data Provisioning Agent is configured in your SAP Datasphere tenant (see [Preparing Data Provisioning Agent Connectivity](#)).
- Create a Cloud Data Integration connection to your SAP Analytics Cloud tenant, using OAuth 2.0 authentication (see [Cloud Data Integration Connections](#)).

See the blog [Using the SAC Data Export API with SAP Datasphere and Data Intelligence](#)  for more information.

Load SAP Analytics Cloud Planning Data to SAP Datasphere

You can load your SAP Analytics Cloud planning data to SAP Datasphere via your Cloud Data Integration connection:

- As a remote table (see [Import Remote Tables](#)).
- Via a data flow (see [Creating a Data Flow](#)).

Combine Actuals and Planning Data in an SAP Datasphere View

You can combine the planning data loaded from SAP Analytics Cloud with your actuals data (or any other appropriate data) in a view and then expose the view to allow it to be consumed in an SAP Analytics Cloud story via a live connection (see [Consume Data in SAP Analytics Cloud via a Live Connection \[page 7\]](#)).

1.3 Consume Data in Microsoft Excel via an SAP Add-In

You can create a live connection from SAP Analytics Cloud to SAP Datasphere and consume data exposed as analytic models and perspectives in Microsoft Excel, via the SAP Analytics Cloud, add-in for Microsoft Excel.

Prerequisites

To consume data exposed by SAP Datasphere in Microsoft Excel, you must have a scoped role that grants you access to a space with the following privileges:

- [Data Warehouse Consumption](#) (-R- - - - -) - To access data exposed by SAP Datasphere spaces.
- [Space Files](#) (-R- - - - -) - To access objects in spaces.

The [DW Consumer](#) role template grants these privileges, and is intended for users who will only consume data and do not need to connect to SAP Datasphere itself. All other standard SAP Datasphere role templates also grant these privileges. For more information, see [Privileges and Permissions](#) and [Standard Roles Delivered with SAP Datasphere](#).

📘 Note

If data access controls are applied, then the data you can consume is filtered based on your user id (see [Securing Data with Data Access Controls](#)).

You must, in addition:

- Be an SAP Analytics Cloud user with the role [BI Content Creator](#) or another role providing equivalent privileges.
- Create or have access to an SAP Analytics Cloud live data connection to your SAP Datasphere tenant (see [Live Data Connections to SAP Datasphere](#) in the *SAP Analytics Cloud* documentation).

📘 Note

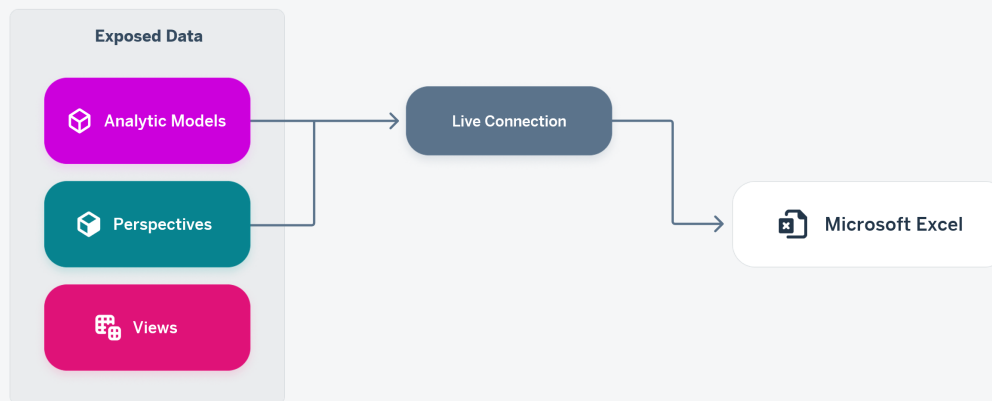
The workflow of consuming views with a semantic usage of [Analytical Dataset](#) in SAP Analytics Cloud and Microsoft Excel via live connection is now deprecated. We recommend that you migrate your analytical datasets to the new [Fact](#) semantic usage and expose your view data via analytic models (see [Analytical Datasets \(Deprecated\)](#)).

- Install the add-in (see [Deploying the Add-In](#) in the *SAP Analytics Cloud, add-in for Microsoft Excel* documentation).

Note

This topic focuses on the SAP Analytics Cloud, add-in for Microsoft Excel. You can also consume data exposed as perspectives and views in the SAP Analysis for Microsoft Office add-in (see the [SAP Analysis for Microsoft Office](#) documentation). Data exposed as analytic models cannot be consumed in the SAP Analysis for Microsoft Office add-in.

Consume Data in Microsoft Excel



Procedure

1. Open your Excel desktop app (or Excel for the Web) and create a new blank workbook.
2. Click the [SAP Analytics Cloud](#) Ribbon tab, and click [Sign In](#) to open the [SAP Analytics Cloud](#) panel.
3. Enter the URL for your SAP Analytics Cloud tenant, and click [Go](#).
4. Enter your SAP Analytics Cloud user name and password, and click [Log On](#) to connect to the tenant.
5. Click [Add Table](#), select your SAP Datasphere connection and space, then select your object, and click [Open](#).

Note

The workflow of consuming views with a semantic usage of [Analytical Dataset](#) in SAP Analytics Cloud and Microsoft Excel via live connection is now deprecated. We recommend that you migrate your analytical datasets to the new [Fact](#) semantic usage and expose your view data via analytic models (see [Analytical Datasets \(Deprecated\)](#)).

6. Select measures and dimensions as appropriate and work with your data as usual.

For detailed information about working in Excel, see [Analyze Your Data](#) in the *SAP Analytics Cloud, add-in for Microsoft Excel* documentation.

See also the blog [Access SAP Datasphere with SAP's Microsoft Excel add-ins](#) (published May 2022).

1.4 Consume Data in Power BI and Other Clients, Tools, and Apps via an OData Service

You can consume data exposed as views in Microsoft Power BI and other third-party clients, tools, and apps via the OData API. You can connect Power BI to SAP Datasphere via an OData service by way of a custom connector or a blank query and consume views that are exposed for consumption.

Prerequisites

To consume data exposed by SAP Datasphere via the OData API, you must have a scoped role that grants you access to a space with the following privileges:

- [Data Warehouse Consumption](#) (-R- - - - -) - To access data exposed by SAP Datasphere spaces.
- [Space Files](#) (-R- - - - -) - To access objects in spaces.

The [DW Consumer](#) role template grants these privileges, and is intended for users who will only consume data and do not need to connect to SAP Datasphere itself. All other standard SAP Datasphere role templates also grant these privileges. For more information, see [Privileges and Permissions](#) and [Standard Roles Delivered with SAP Datasphere](#).

Note

If data access controls are applied, then the data you can consume is filtered based on your user id (see [Securing Data with Data Access Controls](#)).

You must, in addition:

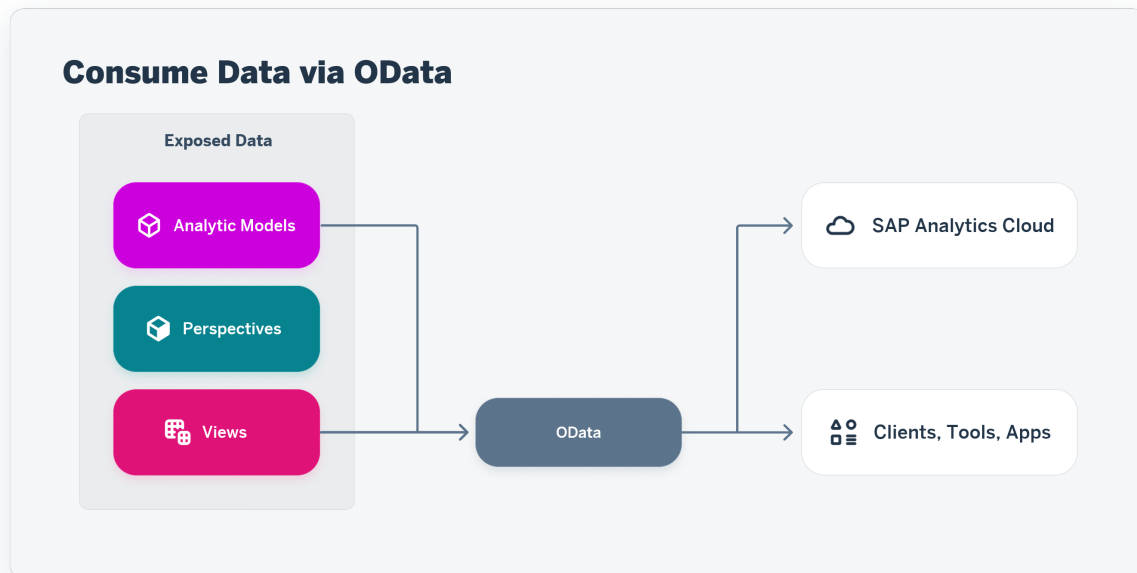
- Obtain the following parameters for an OAuth client defined in your SAP Datasphere tenant with [Purpose](#) set to [Interactive Usage](#) or [Technical User](#):

Standard OAuth2 Authorization Flow	OAuth2SAMLBearer Principal Propagation Flow
• Client ID • Secret • Authorization URL (not required for clients with a Technical User purpose) • Token URL Users of a client with a Technical User purpose (which includes its own privileges and permissions) can then access their resource directly. Users of clients with other purposes must manually authenticate against the IDP in order to generate the authorization code before continuing with the remaining OAuth2.0 steps.	• Client ID • Secret • OAuth2SAML Token URL • OAuth2SAML Audience Users of a client with a Technical User purpose can then access their resource directly. Users of clients with other purposes must authenticate with their third-party app, which has a trusted relationship with the IDP, and do not need to re-authenticate (see Add a Trusted Identity Provider). See also the blog Integrating with SAP Datasphere Consumption APIs using SAML Bearer Assertion  (published March 2024).

Note

Consuming exposed data in third-party clients, tools, and apps via an OData service requires a three-legged OAuth2.0 flow with type `authorization_code`.

For information about OAuth clients, see [Create OAuth2.0 Clients to Authenticate Against SAP Datasphere](#).



Procedure

See the following blogs for more information:

- [SAP Datasphere: OData Connector for Power BI](#) (published October 2022)
- [Connecting SAP Datasphere OData API with Power BI via a Blank Query](#) (published September 2022)

For detailed information about working with OData, see [Consuming Data via the OData API \[page 20\]](#).

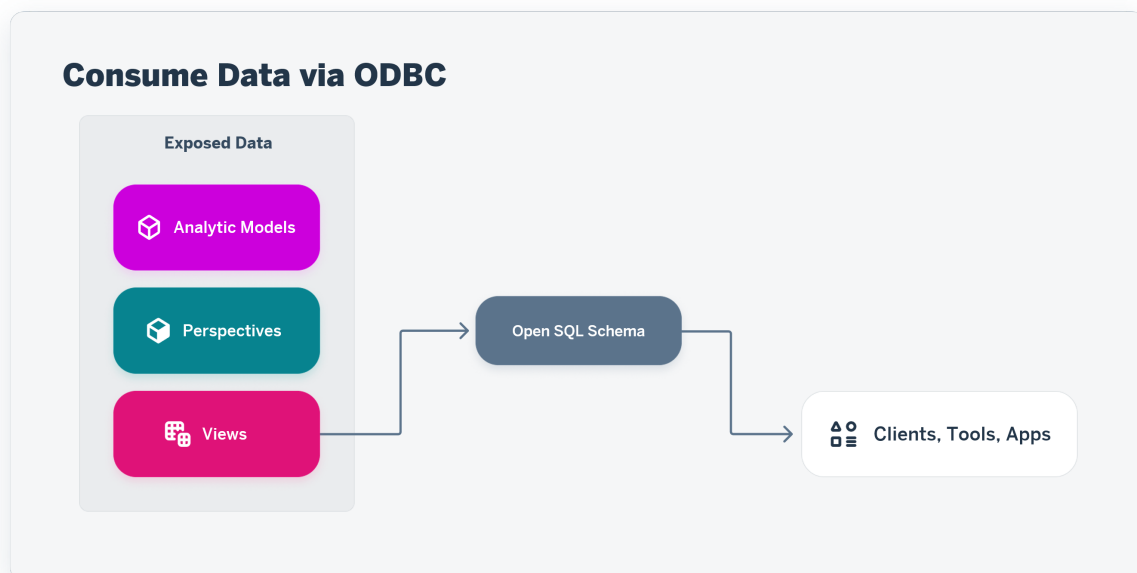
1.5 Consume Data in Power BI and Other Clients, Tools, and Apps via ODBC/JDBC

You can consume data exposed as views in Microsoft Power BI and other third-party clients, tools, and apps via an Open SQL schema and ODBC/JDBC.

Prerequisites

You must:

- Request a space administrator or integrator to provide you with access to a database user/Open SQL schema associated with the SAP Datasphere space exposing the data (see [Create a Database User](#)).
- Request an administrator to add the IP address of your tool to the IP allowlist (see [Manage IP Allowlist](#)).
- Install the SAP HANA CLIENT 2.0 (see [Download and Install the SAP HANA Client](#) in the *SAP HANA Cloud* documentation).
- Create an ODBC or JDBC connection to SAP Datasphere using your database user credentials (see [Connect to SAP HANA Cloud via ODBC](#) or [Connect to SAP HANA Cloud via JDBC](#) in the *SAP HANA Cloud* documentation).



Procedure

You can connect Power BI to SAP Datasphere via an Open SQL schema, and consume perspectives and views that are exposed for consumption.

See the blog [How to Consume SAP Datasphere in Power BI](#)  for more information.

1.6 Consuming Data via the OData API

You can connect to the OData API and consume data exposed as views or analytic models in SAP Analytics Cloud and other clients, tools, and apps that are capable of accessing an OData service and authenticating via an OAuth client.

This topic contains the following sections:

- [Prerequisites \[page 20\]](#)
- [Introduction to the OData APIs \[page 22\]](#)
- [Generate OData Requests from SAP Datasphere Editors \[page 23\]](#)
- [List Spaces and Assets Exposed for Consumption \[page 24\]](#)
- [Retrieve Analytical Data \(Analytic Models\) \[page 24\]](#)
- [Retrieve Relational Data \(Facts, Dimensions and Other Views\) \[page 25\]](#)
- [Retrieve Asset Service URL \[page 26\]](#)
- [Retrieve Asset Metadata \[page 26\]](#)
- [Use the OData \\$batch Syntax \[page 26\]](#)
- [Set Values for Input Parameters / Variables \[page 27\]](#)
- [Use Query Parameters \[page 28\]](#)
- [Pagination \[page 30\]](#)
- [OData Annotation Limitations \[page 31\]](#)
- [API Rate Limiting \[page 31\]](#)

Prerequisites

To consume data exposed by SAP Datasphere via the OData API, you must have a scoped role granting access to the appropriate spaces with the following privileges:

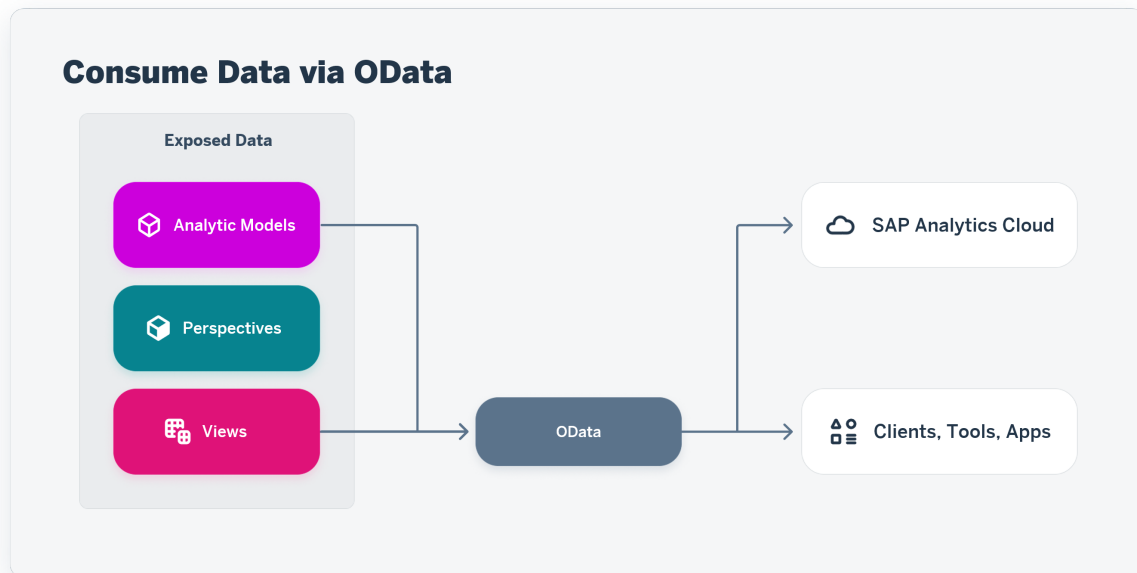
- [Data Warehouse Consumption](#) (-R- - - - -) - To access data exposed by SAP Datasphere spaces.
- [Space Files](#) (-R- - - - -) - To access objects in spaces.

You must, in addition obtain access to an appropriate OAuth client:

OAuth Purpose	Interactive Usage	Technical User	API Access
Authorization Grant	Authorization Code Three- Legged (User-Client-Server): Manually propagate authorization to another service.	Client Credentials Two-Legged (Client-Server): Service to service communication via APIs.	SAML2.0 Bearer Access: Data Export Service SAML Assertion: Seamlessly propagate authorization to another service.
Authentication	Manually authenticate to generate the authorization code.	None required. The OAuth client contains the necessary credentials and privileges. When apply data access controls, use the user ID associated with the OAuth client (see Securing Data with Data Access Controls). See also the blog Using a Technical User in SAP Datasphere Consumption APIs (Client Credentials) (published September 2025).	Authenticate with your third-party app, which has a trusted relationship with the IDP (see Add a Trusted Identity Provider). See also the blog Integrating with SAP Datasphere Consumption APIs using SAML Bearer Assertion (published March 2024).
Parameters	• Client ID • Secret • Authorization URL • Token URL	• Client ID • Secret • Token URL	• Client ID • Secret • OAuth2SAML Token URL • OAuth2SAML Audience

For more information, see [Create OAuth2.0 Clients to Authenticate Against SAP Datasphere](#).

Introduction to the OData APIs



Compose your OData request based on your SAP Datasphere tenant URL in the following format:

```
GET https://<tenant_url>/api/v1/datasphere/consumption/<request>
```

Note

The former URL format, GET `https://<tenant_url>/api/v1/dwc/consumption/<request>` is deprecated and support will be discontinued in a future release.

Note

If your SAP Datasphere tenant is included in an SAP Business Data Cloud formation or is used as data storage for planning by an SAP Analytics Cloud tenant, then this URL is not accessible via Web browser without an explicit OAuth authentication.

See also [Use the OData \\$batch Syntax \[page 26\]](#).

Requests return, by default, the first 10 results and for retrieving more or other pages you should make use of the `$top` and `$skip` parameters.

The API specification is available at the [SAP Business Accelerator Hub](#).

See also the blog [SAP Datasphere: Analytical and Relational OData APIs](#) (published January 2024).

Note

Only the parts of the [OData v4 specification](#) described in this document are supported. Write operations are not supported.

Generate OData Requests from SAP Datasphere Editors

If you have access to the SAP Datasphere application, you can generate requests to access data and metadata directly from the relevant asset editor and either copy it to the clipboard or download it as an OpenAPI request.

To access SAP Datasphere and generate an OData request in an object editor, you must have a scoped role that grants you access to a space with the following privileges:

- [Data Warehouse General](#) (-R- - - - -) - To access SAP Datasphere.
- [Space Files](#) (-R- - - - -) - To access objects in spaces.
- [Data Warehouse Data Builder](#) (CRUD- - - -) - To open views and analytic models.

The [DW Viewer](#) role template, for example, grants these privileges. For more information, see [Privileges and Permissions](#) and [Standard Roles Delivered with SAP Datasphere](#).

1. Open your analytic model or view in its editor. See:

- [Creating an Analytic Model](#)
- [Creating a Graphical View](#)
- [Creating an SQL View](#)

Note

Only views with the [Expose for Consumption](#) switch enabled can generate an OData request. Views and analytic models must be deployed before their data can be consumed.

2. Click  ([Generate OData Request](#)) to open the [Generate OData Request](#) dialog.
3. Complete the dialog as appropriate:

Property	Description
Request	Select from: • Data [default] - Retrieve asset data. See: • Retrieve Analytical Data (Analytic Models) [page 24] • Retrieve Relational Data (Facts, Dimensions and Other Views) [page 25] • Metadata - Retrieve asset XML definition (see Retrieve Asset Metadata [page 26]). • Service - Retrieve asset service URL (see Retrieve Asset Service URL [page 26]).
Variables/ Input Parameters	Enter a value for each variable (analytic model) or input parameter (view) required by the asset (see Set Values for Input Parameters / Variables [page 27]). You can use the Add button to enter additional values for variables supporting multiple values.
OData Parameters	Add and set values for any required OData parameters (see Use Query Parameters [page 28]).

Note

When configuring variables and parameters, you should always avoid disclosing sensitive data in the generated URL.

4. [optional] Click [Preview](#) to preview up to 100 records of data.
You can click [Copy Preview Data](#) to copy the data from the preview.
5. Click [Copy URL](#) or [Download OpenAPI Request](#) to obtain the request in the format you need.

6. Click [Close](#) to close the dialog and return to the editor.

List Spaces and Assets Exposed for Consumption

Use the `catalog` request to list spaces and assets that are exposed for consumption.

To list all the spaces you have access to, enter:

```
GET https://<tenant_url>/api/v1/datasphere/consumption/catalog/spaces
```

To list all the assets you have access to, enter:

```
GET https://<tenant_url>/api/v1/datasphere/consumption/catalog/assets
```

To list all the assets exposed in a particular space, enter:

```
GET https://<tenant_url>/api/v1/datasphere/consumption/catalog/spaces('<space_id>')/assets
```

To get access to the description of a particular asset, including links to its metadata and data, enter:

```
GET https://<tenant_url>/api/v1/datasphere/consumption/catalog/spaces('<space_id>')/assets('<asset_id>')
```

Retrieve Analytical Data (Analytic Models)

To retrieve the analytical data for an asset, enter:

```
GET https://<tenant_url>/api/v1/datasphere/consumption/analytical/<space_id>/<asset_id>/<asset_id>[(<asset_params>)/Set][?<query_params>]
```

Note

Analytic models support only `consumption/analytical` requests.

By default, all measures and attributes are returned up to a maximum of 50,000 records.

You can use the `$select` OData query parameter to aggregate the results by specifying at least one measure or attribute.

For example, to aggregate the measure `Sales` by the attribute `Country` you would enter:

```
GET https://<tenant_url>/api/v1/datasphere/consumption/analytical/<space_id>/<asset_id>/<asset_id>?$select=Sales, Country
```

The `$select` OData query parameter allows you to perform aggregations of measure values or obtain lists of attribute/dimension values. If you specify:

Measures	Attributes	Output
1+	0	Total aggregate values for each measure. For example, <code>?\$select=Sales</code> returns the total value of sales.
1+	1	Aggregate values for each measure for each unique attribute value. For example, <code>?\$select=Sales,Country</code> returns sales per country.
1+	2+	Aggregate values for each measure for each unique combination of attribute values. For example, <code>?\$select=Sales,Country,Product</code> returns sales per country per product.
0	1	List of unique attribute values. For example, <code>?\$select=Country</code> returns the list of countries.
0	2+	List of unique combinations of attribute values. For example, <code>?\$select=Country,Product</code> returns the list of unique country-product combinations.

You can also use OData query parameters to restrict the data you retrieve and control pagination (see [Use Query Parameters \[page 28\]](#)).

If the asset requires you to set a value for one or more `<asset_params>`, see [Set Values for Input Parameters / Variables \[page 27\]](#).

Retrieve Relational Data (Facts, Dimensions and Other Views)

To retrieve the relational data for an asset, enter:

```
GET https://<tenant_url>/api/v1/datasphere/consumption/relational/<space_id>/<asset_id>/<asset_id>[(<asset_params>)/Set][?<query_params>]
```

Note

Views with a semantic usage of *Fact* and most other views support only relational requests. Views with the deprecated semantic usage of *Analytical Dataset* support both relational and analytical requests.

By default, all columns and records are returned up to a maximum of 50,000 records.

You can use OData query parameters to restrict the data you retrieve and control pagination (see [Use Query Parameters \[page 28\]](#)).

If the asset requires you to set a value for one or more `<asset_params>`, see [Set Values for Input Parameters / Variables \[page 27\]](#).

Retrieve Asset Service URL

To retrieve the service URL for an analytic model, enter:

```
GET https://<tenant_url>/api/v1/datasphere/consumption/analytical/<space_id>/<asset_id>
```

To retrieve the service URL for a view, enter:

```
GET https://<tenant_url>/api/v1/datasphere/consumption/relational/<space_id>/<asset_id>
```

Retrieve Asset Metadata

To retrieve the metadata for an asset, including its measures, attributes, and variables, enter:

```
GET https://<tenant_url>/api/v1/datasphere/consumption/relational/<space_id>/<asset_id>/$metadata
```

For information about the semantic types and the annotations associated with columns, see:

- [Specify Semantic Types for Measures and Attributes](#)
- [SAP/odata-vocabularies/vocabularies/Common.md](#)  on GitHub

Use the OData \$batch Syntax

Certain data consumers, including SAP Build and SAP Fiori Elements Apps, require the use of the OData \$batch syntax.

Note

SAP Datasphere restricts the contents of the body of a \$batch request to a single GET request. Additional requests are not supported.

You must use a POST request with the \$batch syntax:

```
POST https://<tenant_url>/api/v1/datasphere/<request>/$batch
```

You can use either a Multipart or JSON format for defining the GET request:

- Multipart:

```
--batch_1
Content-Type: application/http
GET /<asset_id>[?<query_params>] HTTP/1.1
odata-version: 4.0
Accept: application/json
--batch_1--
```

Note

We do not support the Multipart format for requests to assets with input parameters / variables. In this situation, you must use the JSON format.

- JSON:

```
{  "requests":  [    {      "id": "1",      "method": "GET",      "url": "<asset_id>[(<asset_params>)/Set][?<query_params>]",      "headers": {        "odata-version": "4.0",        "Accept": "application/json"      }    },  ]}
```

Note

When sending a \$metadata request, the response is in base64 encoding and must be decoded on the client side.

Set Values for Input Parameters / Variables

Some assets exposed for consumption may allow you to set values to filter or otherwise modify the data you consume.

To set values, append the following syntax (where `<asset_param>` is the technical name of your parameter) directly after your `<asset_id>`:

```
(<asset_param>=<val>[, ...])/Set
```

Note

Text values must be enclosed in single quotes. Text values that contain single or double quotes are not supported.

Variables and input parameters supporting only single values must be entered as follows:

Value Type	Example Syntax
Single Numeric Value	(Sales_Amount=100)/Set
Single Text Value	(Product_Name='Product A')/Set
Single Date Value	(Date=2023-01-02)/Set

Variables supporting multiple values require the use of a two-character operator (which can be in uppercase or lowercase), with the operator/values expression string enclosed in single quotes (and any referenced text values enclosed, in turn, in double quotes):

Operator	Example
eq / ne (Equal to / Not equal to)	(Product_ID='eq 8013311')/Set
gt / ge (Greater than / Greater than or equal to)	(Date='ge 2023-01-02')/Set
lt / le (Less than / Less than or equal to)	(Price='lt 1000')/Set
bt (Between the lower and upper bound values)	(Sales_Date='bt 2023-01-01,2023-01-31')/Set
in (In the comma-separated list of values)	(Product_Code='in "CT-100", "PM-255", "ZH-133"')/Set

Some input parameters and variables have default values defined. If your input parameter or variable has:

- No default value:
 - You must set a value for it.
- A default value
 - If you do not set a value for it, the default value will be used.
 - You can set a value and your value will be used in place of the default value.

For example, **Sales_View** in the **SALES** space has two input parameters:

- **Region** - which has no default value.
- **Category** - which has a default value of **Footwear**.

To view sales of US footwear, enter:

```
GET https://<tenant_url>/api/v1/datasphere/consumption/relational/SALES/Sales_View/Sales_View(Region='US')/Set
```

To view sales of US apparel, enter:

```
GET https://<tenant_url>/api/v1/datasphere/consumption/relational/SALES/Sales_View/Sales_View(Region='US',Category='Apparel')/Set
```

For more information, see [Create an Input Parameter in a Graphical View](#) and [Create a Variable in an Analytic Model](#).

Use Query Parameters

You can use the following OData query parameters to control the data you retrieve:

Parameter	Description
\$select	Requests the service to return only the specified properties (see System Query Option \$select in the <i>OData Version 4.0 Specification</i>). For example, to retrieve only the columns Partner_ID and Value , append the syntax: ?\$select=Partner_ID, Value

Parameter	Description
\$filter	Requests the service to restrict returned results according to the provided dimension or measure criteria: • eq - Equals • ne - Not equal • gt - Greater Than • ge - Greater Than or Equal To • lt - Less Than • le - Less Than or Equal To • startswith(<field>, '<value>') - String value starts with • endswith(<field>, '<value>') - String value ends with • contains(<field>, '<value>') - String value contains • and - And • or - Or • not - Not • () - Grouping For example, to retrieve only records: • Where dimension Partner_ID is equal to 100000005, append the syntax: <pre>?\$filter=Partner_ID eq 100000005</pre> • Where dimension Date is between 2023-01-01 and 2023-12-31 inclusive, append the following syntax: <pre>?\$filter=Date ge 2023-01-01 and Date le 2023-12-31</pre> • Where Product begins with TV* and Country is US, append the following syntax: <pre>?\$filter=startswith(Product, 'TV') and Country eq US</pre> • Where dimension Partner_ID is equal to 100000005 and measure Value is greater than one million, append the syntax: <pre>?\$filter=Partner_ID eq 100000005 and Value gt 1000000</pre> Note Values containing single quotes cannot be used. Filtering on dimensions is performed before the cube is generated, while filtering on aggregated measures must be performed afterwards, and so is necessarily more costly in terms of performance.
\$orderby	Specifies the order in which items are returned (see System Query Option \$orderby in the <i>OData Version 4.0 Specification</i>). For example, to order records by descending order of Value, append the syntax: <pre>?\$orderby=Value desc</pre>

Parameter	Description
<code>\$count=true</code>	Returns the count of the number of records in the dataset (see System Query Option \$count in the <i>OData Version 4.0 Specification</i>). Note Not supported for analytical requests.
<code>\$top=<n></code>	Limits the number of records returned from the dataset (see System Query Option \$top in the <i>OData Version 4.0 Specification</i>). For example, to retrieve only the top 5 records by descending order of Value, append the syntax: <pre>?\$orderby=Value desc&\$top=5</pre>
<code>\$skip=<n></code>	Excludes the first <code><n></code> items of the selected collection (see System Query Option \$skip in the <i>OData Version 4.0 Specification</i>). For example, to retrieve only records 6-10 by descending order of Value, append the syntax: <pre>?\$orderby=Value desc&\$skip=5&\$top=5</pre>
<code>sap-language=<xx></code>	Returns data in the language specified by the two-character ISO 639 language code <code><xx></code>. Data in this language must be available via a text entity (Create a Text Entity for Attribute Translation). If no data is available in the specified language, then the attribute will be empty. For example, to retrieve records in German, append the syntax: <pre>?sap-language=de</pre> If this parameter is not specified, then data is returned in: - The language specified in the Accept - Language request header, or if not set, - The Language specified in your settings (see Changing SAP Datasphere Settings), or if not set, - English.

Note

Query parameters have the following limitations if any source in the asset's lineage is not replicated to SAP Datasphere:

- `$count`, `$top` and `$skip` are not supported.
- `$filter` only supports `eq`, `and`, `or`, and `()`.

To view the lineage of your asset, see [Impact and Lineage Analysis](#).

Pagination

Where the request will not fit into a single response, a link annotated as `@odata.nextLink` is provided at the end of the response to allow you to move to the next page. If client side pagination is applied (or at the final page) there will be no next link.

You can use the `$count=true` OData parameter to obtain the number of records in your request, and enable client side pagination with `$top` and `$skip`.

Request Service	Default Page Size	Maximum Page Size
catalog	500 records	500 records
relational	50KB (up to a maximum of 50,000 records)	100KB (up to a maximum of 100,000 records)
analytical	50KB (up to a maximum of 50,000 records or 1 million cells)	100KB (up to a maximum of 100,000 records or 1 million cells)

OData Annotation Limitations

Annotations in the metadata of a specific service might restrict the query options. Some of the CDS data types are partially or not supported by the SAP Datasphere Consumption API.

Data Type	Writable	Sortable	Filterable	Overwritten EDM Type
cds.hana.ST_GEOMETRY	✗	✗	✓	Edm.String
cds.hana.ST_POINT	✗	✗	✓	Edm.String
cds.Binary		✗	✗	-
cds.LargeBinary	✗	✗	✗	-
cds.hana.BINARY	✗	✗	✗	-
cds.LargeString	✓	✗	✓	-
cds.UUID	✓	✓	✗	-

API Rate Limiting

For information about SAP's general API controls and restricted use cases, see the [SAP API Policy](#).

Authenticated requests are associated either with the authenticated username, the OAuth client ID, or the tenant ID. Unauthenticated requests are associated with the originating IP address, and not the user.

Requests are limited to approximately 300 per user per minute. If you exceed the limit, you will receive the HTTP 429 Too Many Requests response status code and can review the following request response headers for further information:

- `X-Ratelimit-Limit` - Rate limit per user per minute.
- `X-Ratelimit-Remaining` - Remaining number of requests for the current timeframe for the current user.
- `X-Ratelimit-Reset` - Time in seconds until the rate limit is reset to the defined limit.
- `Retry-After` - Time in seconds the user agent should wait before making a follow-up request.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

© 2026 SAP SE or an SAP affiliate company. 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 SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company 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.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.

