SAP SuccessFactors

PUBLIC

Document Version: 1H 2022 – 2022-06-17

SAP SuccessFactors HXM Suite OData API: Developer Guide (V4)

THE BEST RUN SAP

Content

1 About the OData API Developer Guide (V4)

The Open Data Protocol (OData) is a standardized protocol for consuming REST APIs. SAP SuccessFactors provides a variety of OData APIs for customers to build their extensions and integrations. In this guide, you'll learn how to work with OData V4 APIs in SAP SuccessFactors HXM Suite, including authentication, common operations, access limits, response codes, and more.

New and Improved Features in OData V4

> **i** Note
>
> The following features are supported in the standard OData V4 protocol. For individual OData V4 offerings in SAP SuccessFactors, the capabilities vary depending on the implementation of each module. Refer to the API references for more information.

Delta Support

Delta support allows clients to query services and receive just the set of deltas - changes - from a previous state. Designed for scalability, this hypermedia-driven model gives the service ultimate control over how changes are identified while providing a simple, well-defined model to the client.

Entity Data Model Enhancements

OData defines a consumer-oriented Entity Data Model that helps general-purpose clients understand how to interact with an OData service. OData V4 enhances the entity model, adding support for containment, singletons, enums, and type definitions. Date/Time data types have been reworked to include separate Date, TimeOfDay, Duration, and DateTimeOffset data types. Complex types now support inheritance and navigation properties. Relationships, a key part of the entity model, are simplified.

Improved Queryability

In today's device-centric world, bringing back the right subset of data is key to reducing round-trips, payload, and footprint. OData V4 includes syntax for applying filters, sorts, and selects against expanded properties, allowing clients to push processing to the service and more precisely specify the desired set of data to be retrieved. Full-text search functionality, recursive queries, and the use of user-defined functions in predicates all help clients control the data returned from the service.

Related Information

This guide provides information specific to the latest version (V4) of OData APIs in SAP SuccessFactors HXM Suite. To fully understand how OData works in general or how OData V2 works in SAP SuccessFactors, refer to the following documentation:

Resource	Description
SAP SuccessFactors HXM Suite OData API: Reference Guide (V4)	Contains API references of OData V4 service groups in SAP SuccessFactors HXM Suite, including service group overview and use cases.
SAP SuccessFactors HXM Suite OData API: Developer Guide	A developer guide to the OData V2 framework, including permissions, authentication, common OData operations, MDF OData APIs, and more.
odata.org	The official OData website where you can find more detailed specifications about the OData standard.
SAP API Business Hub	SAP SuccessFactors publishes customer-facing OData APIs on the SAP API Business Hub. You can find all technical details of our public APIs and run tests in the sandbox system.

1.1 List of SAP SuccessFactors API Servers

Learn about the API servers of your company instance and how to construct the endpoint URLs.

Endpoint URL Patterns

> **!** Restriction
>
> We don't support IP addresses in URLs as part of our reference architecture. Use domain names instead. If you think you have a special case that requires IP addresses instead of domain names, contact Product Support.

Protocol	URL Pattern	Endpoint Example
OData v2	`/odata/v2/`	`https://api17.sapsf.com/odata/v2/`
OData v4	`/odatav4/`	`https://api17.sapsf.com/odatav4/`
SFAPI	`/sfapi/v1/soap`	`https://api17.sapsf.com/sfapi/v1/soap`
WSDL	`/sfapi/v1/soap?wsdl`	`https://api17.sapsf.com/sfapi/v1/soap?wsdl`

API Servers

Here's a list of API servers for SAP SuccessFactors data centers. Use search and filter to find the API server for your company. A data center can have synonyms from its legacy and next generation data center names, indicated by a numeric value. For example, DC17 and DC60 are synonyms of the Toronto data center. The corresponding URLs also point to the same data center. You can use either URL to access APIs.

To view the timezone information of an API server, go to your company login page or open your account on the header bar after login, and choose *Show version information*.

> **i Note**
>
> SAP SuccessFactors revised its data center numbering as part of its Next Generation Cloud Delivery Platform. While it is leading practice to update a domain name or URL to the latest domain name, the domain names that you migrated **from** continue to work. For example, if the **Next Generation Cloud Delivery Platform** migrated you from a URL with DC17 in the domain name to a URL with DC60 in the domain name, the DC17 URL, `https://api17.sapsf.com`, continues to point to the same Toronto data center.

Legacy Numeric Name	Next Generation Numeric Name (Planned)	Environment	Location	API Server
DC2	DC56	Production	Amsterdam, The Netherlands	`https://api2.successfactors.eu/`
DC2	DC56	SalesDemo	Amsterdam, The Netherlands	`https://apisalesdemo2.successfactors.eu/`
DC2	DC56	Preview	Amsterdam, The Netherlands	`https://api2preview.sapsf.eu/`
DC4	DC48	Production	Chandler, Arizona, US	`https://api4.successfactors.com/`
DC4	DC48	SalesDemo	Chandler, Arizona, US	`https://apisalesdemo4.successfactors.com/`
DC4	DC48	Preview	Chandler, Arizona, US	`https://api4preview.sapsf.com/`
DC8	DC48	Production	Ashburn, Virginia, US	`https://api8.successfactors.com/`
DC8	DC48	SalesDemo	Ashburn, Virginia, US	`https://apisalesdemo8.successfactors.com/`
DC8	DC48	Preview	Ashburn, Virginia, US	`https://api8preview.sapsf.com/`
DC10	DC66	Production	Sydney, Australia	`https://api10.successfactors.com/`
DC10	DC66	Preview	Sydney, Australia	`https://api10preview.sapsf.com/`
DC12	DC26	Production	Rot, Germany	`https://api012.successfactors.eu/`

Legacy Numeric Name	Next Generation Numeric Name (Planned)	Environment	Location	API Server
DC12	DC26	Preview	Rot, Germany	https://api12preview.sapsf.eu/
DC15	DC64	Production	Shanghai, China	https://api15.sapsf.cn/
DC17	DC60	Preview	Toronto, Canada	https://api17preview.sapsf.com/
DC17	DC60	Production	Toronto, Canada	https://api17.sapsf.com/
DC18	DC28	Preview	Moscow, Russia	https://api18preview.sapsf.com/
DC18	DC28	Production	Moscow, Russia	https://api18.sapsf.com/
DC19	DC62	Preview	Sao Paulo, Brazil	https://api19preview.sapsf.com/
DC19	DC62	Production	Sao Paulo, Brazil	https://api19.sapsf.com/
DC22	DC22	Preview	Dubai, UAE	https://api22preview.sapsf.com/
DC22	DC22	Production	Dubai, UAE	https://api22.sapsf.com/
DC23	DC23	Preview	Riyadh, Saudi Arabia	https://api23preview.sapsf.com/
DC23	DC23	Production	Riyadh, Saudi Arabia	https://api23.sapsf.com/
DC41	DC41	Preview	US East (Microsoft Azure)	https://api41preview.sapsf.com
DC41	DC41	Production	US East (Microsoft Azure)	https://api41.sapsf.com
DC44	DC52	Preview	Singapore	https://api44preview.sapsf.com/
DC44	DC52	Production	Singapore	https://api44.sapsf.com/
DC47	DC47	Preview	Canada Central (Microsoft Azure)	https://api47preview.sapsf.com/
DC47	DC47	Production	Canada Central (Microsoft Azure)	https://api47.sapsf.com/
-	DC50	Preview	Aisa Northeast, Tokyo (Google Cloud Platform)	https://api50.sapsf.com
-	DC50	Production	Aisa Northeast, Tokyo (Google Cloud Platform)	https://api50preview.sapsf.com
DC55	DC55	Preview	Europe West 3	https://api55preview.sapsf.eu/
DC55	DC55	Production	Europe West 3	https://api55.sapsf.eu/

1.2 Summary of Differences Between OData V2 and V4

Learn about the differences between OData v2 and v4 protocols in SAP SuccessFactors.

Differences Between OData v2 and v4

Capability	OData v2	OData v4	More Information
Metadata	Supports XML format only	Supports both XML and JSON formats • XML: `/odatav4/Sample.svc/v1/$metadata` • XML: `/odatav4/Sample.svc/v1/$metadata?$format=JSON`	OData V4 Metadata [page 28]
Metadata Scope	Supports single, full, and comma-delimited metadata queries, examples: • Single: `/odata/v2/User/$metadata` • Full: `/odata/v2/User/$metadata` • Comma-delimited: `/odata/v2/User,Picklist/$metadata`	Supports metadata query on service level only.	
Data Type	Supports the following data types: • DateTime • Time • Float	Supports the following data types: • Date • TimeofDay • Single • Enum	
Merge Operation	OData v2 uses HTTP method `POST` and HTTP header `X-HTTP-METHOD: MERGE` to merge records.	In OData v4, you can use the `PATCH` HTTP method to merge records.	
Upsert	Supports SAP-proprietary function import `/odata/v2/upsert` with HTTP method `POST`	Not supported	

Capability	OData v2	OData v4	More Information
Function import and action import	Uses HTTP method `GET` for read operations. Uses HTTP method `POST` for write operations. No action imports.	Uses HTTP method `GET` for function imports (read operations). Uses HTTP method `POST` for function imports (write operations). OData v4 also supports resource-bound functions and actions.	
Entity reference	Supports entity referencing operations with `$link`	`$ref` isn't supported yet.	
Server-side pagination	Supports cursor-based and snapshot-based pagination	Not supported	
Query resources	System query options are supported only at root entity level, for example, `$filter`, `$select`, and `$orderby`.	Supports system query options on aggregated entities: `/odata/v4/User('admin')/reports?$filter=username eq 'abc'` In this example, the filter condition applies to all users whose manager is user 'admin'.	
System query option `$select`	• Applies to root entities only, for example: `/odata/v2/User?$filter=username eq 'abc'` • Nested filters can be used in a one-to-many relationship: `/odata/v2/User?$filter=reports/username eq 'abc'` • Supports custom binary operators	Can be used to query both root and expanded entities: `/odatav4/User?$select=username&$expand=reports($select=username)`	

Capability	OData v2	OData v4	More Information
System query option `$filter`	• Applies to root entities only. • Nested filters can be applied to navigation properties with a one-to-many relationship: `/odata/v2/User?$filter=reports/username eq 'abc'` • Supports customized operators such as `like` and `in` • `datetimeoffset` must be specified in expression before a DateTimeOffset field type: `$filter=lastModified gt datetimeoffset'2020-05-20T00:00:00Z'` • Supports method expression `replace` (v2 only): `$filter=replace(username, 'abc', 'def') eq 'hdef'`	• Can be used to query both root and expanded entities: `/odatav4` • Nested filters can be applied to entities of one-to-many relationship by lambda expressions such as `any` and `all`: `$filter=user=reports/any(m:m/username eq 'abc'` • Customized operators aren't supported. • DateTimeOffset field type can be filtered without `datetimeoffset`: `$filter=lastModified gt '2020-05-20T00:00:00Z'` • Supports method expression `contains` (v4 only): `$filter=contains(username, 'abc')`	

System query option `$expand`	Supports deep expand:
	`/odata/v2/User?$expand=reports,reports/reports`
		Supports deep expand and query options in expanded entities:
		`/odatav4/User?$expand=reports($expand=reports)`
		`/odatav4/User?$expand=reports($filter=username eq 'abc';$select=username)`	

Default response format	ATOM	JSON	

Capability	OData v2	OData v4	More Information
Instance annotations in response	Not supported	Annotations are supported as an extended feature in OData v4. Annotations start with `$`: `{ "@context": "$metadata#Customers", "@com.example.customer.setkind": "VIPs", "value": [{ "@com.example.display.highlight": true, "ID": "ALFKI", "CompanyName@com.example.display.style": { "title": true, "order": 1 }, "CompanyName": "Alfreds Futterkiste", "Orders@com.example.display.style#simple": { "order": 2 } }] }`	

Capability	OData v2	OData v4	More Information

Error response	OData v2 error response example:
	
	{
	  "error": {
	  "code": "BadRequest",
	  "message": {
	  "lang": "en",
	  "value": "Unsupported functionality"
	  }
	}
	
		OData v4 error response example:
		
		{
		  "error": {
		  "code": "BadArgument",
		  "message": "Please check your inputs",
		  "details": [
		  {
		  "code": "NameRequired",
		  "message": "Name must not be empty",
		  "target": "name"
		  },
		  {
		  "code": "PasswordNotMeetPolicy",
		  "message": "Password must have one upper case".
		  "target": "password"
		  }
		  ]
		  }
		}	

Framework-level access limits	No	Yes. See Service Limits [page 35].	

2 Change History

Learn about changes to the documentation for SAP SuccessFactors HXM Suite OData API: Developer Guide (v4) in recent releases.

2H 2021

Type of Change	Description	More Info
Oct 8, 2021		
Added	We added a topic to summarize the major differences of capabilities between OData v2 and OData v4.	Summary of Differences Between OData V2 and V4 [page 7]
Added	We added information about the planned data center names available after the Next Generation Cloud Delivery Platform migration.	List of SAP SuccessFactors API Servers [page 4]

1H 2021

Type of Change	Description	More Info
New	This is the first version of this guide.	

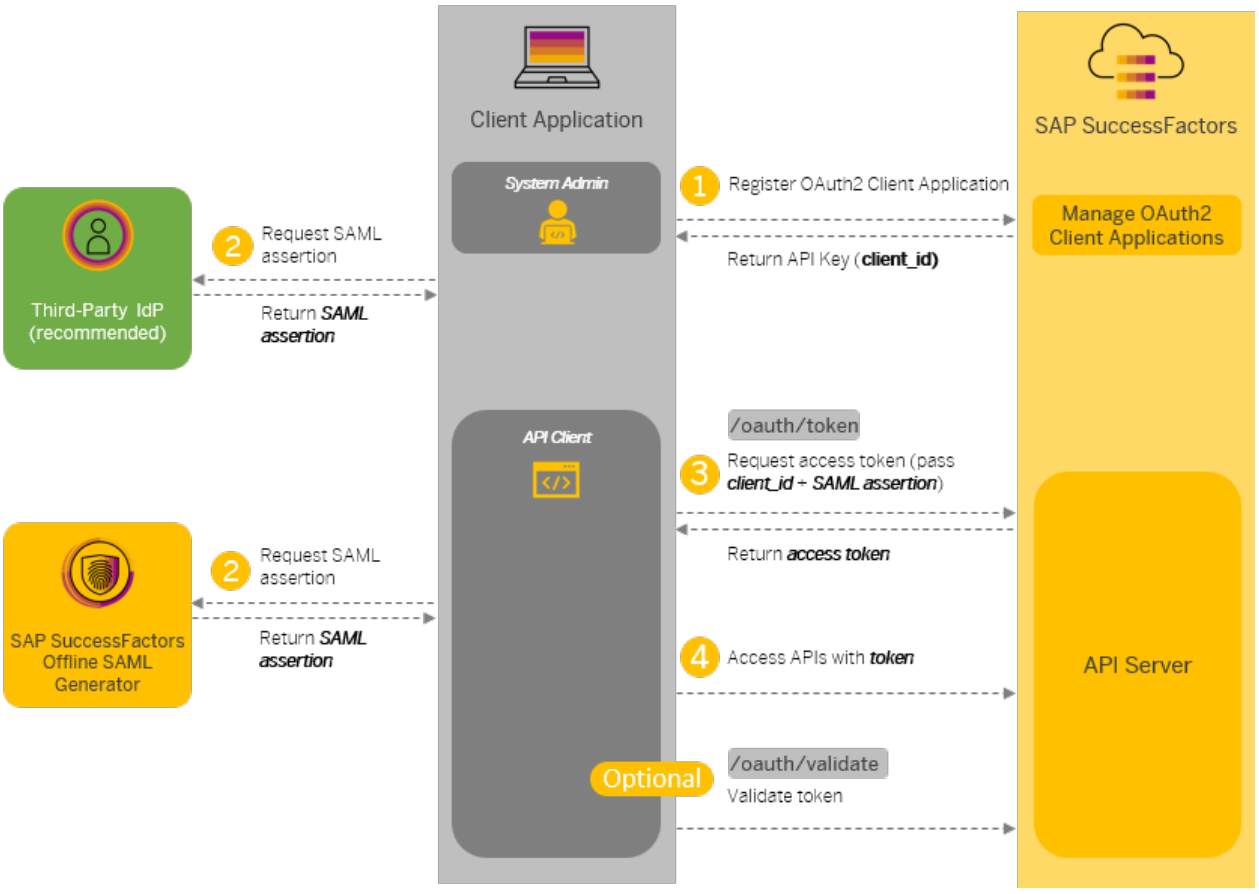
3 Authentication Using OAuth 2.0

Learn how to set up and use OAuth 2.0 for authenticating API users.

SAP SuccessFactors supports OAuth 2.0 to authenticate OData API and SFAPI users. Compared with HTTP Basic Auth, OAuth 2.0 is considered to be more secure in that it doesn't require users to provide their passwords during authentication. With OAuth 2.0, you can also use a third-party identity provider (IDP) for user management and provisioning.

Process Overview

The following diagram explains how OAuth 2.0 works with SAP SuccessFactors.



- Registering Your OAuth2 Client Application [page 14]
- Generating a SAML Assertion [page 21]

- Requesting an Access Token [page 25]
- Viewing the Validity of an Access Token [page 26]
- Follow the documentation of the IdP for requesting SAML assertions. [page 13]

1. Register your client application in SAP SuccessFactors to obtain an API key.
2. Obtain a SAML assertion either from your trusted IdP (recommended) or using the offline SAML generator provided by SAP SuccessFactors.
3. Pass your SAML assertion and API key (in the client_id field) along with other information to generate an OAuth token.
4. Use the generated token to call APIs.
5. (Optional) Check whether your access token has expired or not.

Registering Your OAuth2 Client Application [page 14]
: Register your client application so that you can authenticate API users using OAuth2. After you register an application, you'll get an exclusive API key for your application to access SAP SuccessFactors OData APIs.

Generating a SAML Assertion [page 21]
: Generate a Security Assertion Markup Language (SAML) assertion for requesting an OAuth token. This topic explains how to generate a SAML assertion using the offline tool provided by SAP SuccessFactors.

Requesting an Access Token [page 25]
: With a SAML assertion, you can now call API `oauth/token` to request an access token for authentication with the API server.

Viewing the Validity of an Access Token [page 26]
: Use API `/oauth/validate` to verify if an access token is valid.

3.1 Registering Your OAuth2 Client Application

Register your client application so that you can authenticate API users using OAuth2. After you register an application, you'll get an exclusive API key for your application to access SAP SuccessFactors OData APIs.

Prerequisites

You have the *Manage Integration Tools* > *Manage OAuth2 Client Applications* permission.

Procedure

1. Log into your instance as an administrator.
2. Go to *Admin Center* > *API Center* > *OAuth Configuration for OData* and choose *Register Client Application*. You can also access the tool by searching *Manage OAuth2 Client Applications* in Action Search.
3. On the new OAuth client registration screen, enter the following information:

Option	Description
Company	The name of your company. This value is prefilled based on the instance of the company currently logged in.
Application Name	(Required) A unique name of your OAuth client.
Description	(Optional) A description of your application.
Application URL	(Required) A unique URL of the page that the client wants to display to the end user. The page contains more information about the client application. This is needed for 3-legged OAuth, however it isn't currently supported.
Bind to Users	(Optional) You can enable this option to restrict the access of the application to specific users including business users and technical users. **i Note** A business user in this context is a user who has permissions to call SAP SuccessFactors APIs for integration purposes. A technical user is a system-generated user created for integrating SAP SuccessFactors with other SAP products and solutions. Refer to About Technical User for more information.
User IDs	(Required if you enabled the *Bind to User* option) Enter the user IDs separated by comma. The binding of business users and technical users works as follows: ○ If you don't bind any user to the application, all business users can request OAuth tokens but technical users can't. ○ If you bind both business users and technical users to the application, only these users can request OAuth tokens. ○ If you bind only technical users to the application, these technical users and any business user can request OAuth tokens. ○ If you bind only business users to the application, only these users can request OAuth tokens. **i Note** Contact your system administrator or Product Support if you don't know the technical user ID of your instance.
X.509 Certificate	(Required) The certificate corresponding to the private and public key used in the OAuth 2.0 authentication process. In

Option	Description
	this flow, SAP SuccessFactors require the public key and the client application has the private key. To register a client application, you must install the public key in SAP SuccessFactors. If you supply that certificate, you must use the RSA-SHA1, RSA-SHA2, or MD5 encryption type for authentication. You can obtain an X.509 certificate from a trusted service provider, or you can use a third-party tool to generate a self-signed certificate. If neither option is available, you can generate an X.509 certificate in SAP SuccessFactors. For more information, see the *Related Information* section of this topic. **i Note** For better security, we recommend that you use a self-signed certificate or one from your trusted service provider. In a `.pem` file, the X.509 certificate is a BASE64-encoded string enclosed between `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----`. Enter only the enclosed string without the beginning and ending lines. Otherwise, an error occurs. When you change or regenerate an X.509 certificate for an application, the existing application client configurations are invalidated. This could lead to application failure until you update the configurations with the new certificate information.

4. Choose *Register* to save your registration.

Results

You've successfully registered your client application for OAuth2 authentication. An API key is generated and assigned to your application. You can view the API key by choosing *View* on the registered application list.

You can also edit, disable, and delete an OAuth2 client registration.

Task overview: Authentication Using OAuth 2.0 [page 13]

Related Information

Generating a SAML Assertion [page 21]
Requesting an Access Token [page 25]
Viewing the Validity of an Access Token [page 26]

3.1.1 Creating a Self-Signed X.509 Certificate

You can use tools such as OpenSSL to create a self-signed X.509 certificate.

Prerequisites

We recommend that you use OpenSSL to create the certificate. For Windows users, you can download the tool at https://www.openssl.org . For Mac and Linux users, OpenSSL is available with the native command-line tools such as Terminal.

Context

X.509 certificates are used in many Internet protocols, including TLS/SSL. An X.509 certificate consists of a public key and a private key. The public key contains the identity information, such as a hostname, an organization, or an individual. The public/private key pair is used to establish secure communication between your application and SAP SuccessFactors.

Procedure

1. Go to the OpenSSL library in your command-line tool.

   For Mac and Linux users, you can call OpenSSL directly in the command tool under the default path. For Windows users, the entry point is the openssl binary, located in the installation folder, for example: `C:\Program Files\OpenSSL-Win64\bin\`.
2. Follow the examples below to create an X.509 certificate:
   - Example of creating a 2048-bit SHA256 key:

     ```
     $ openssl req -nodes -x509 -sha256 -newkey rsa:2048 -keyout private.pem -out public.pem
     ```
   - Example of creating a 1024-bit SHA256 key:

     ```
     $ openssl req -nodes -x509 -sha256 -newkey rsa:1024 -keyout private.pem -out public.pem
     ```

- Example of creating a 1024-bit MD5 key:

```
$ openssl req -nodes -x509 -md5 -newkey rsa:1024 -keyout private.pem -out
public.pem
```

> **i Note**
>
> **private.pem** and **public.pem** are the example names of the public/private key pair generated with this command. You can change them to any names of your choice.
>
> SAP SuccessFactors support certificates with SHA1, SHA256, and MD5 encryption types and key lengths from 512 bits to 2048 bits. However, for maximum security, we recommend that you use 2048-bit keys with SHA256 encryption.

3. Enter the following information when prompted:

   Provide at least one of these values to create a certificate.

Option	Description
Country Name	Enter a two-letter country code of the entity to which the certificate is issued. A country code represents a country or a region. Example: **AU**
State or Province Name	Name of state or province of the entity to which the certificate is issued.
Locality Name	Name of locality of the entity to which the certificate is issued.
Organization Name	The entity to which the certificate is issued.
Organization Unit Name	The organization unit of the entity to which the certificate is issued.
Common Name	The hostname or IP address for which the certificate is valid. The common name (CN) represents the hostname of your application. It's technically represented by the commonName field in the X.509 certificate. The common name doesn't include any protocol, port number, or path. For example: **www.bestrun.com**
E-mail Address	Enter your e-mail address.

Results

A public/private key pair is generated and saved to the local drive with the names you specified in the command.

> **⚠ Caution**
>
> Only the public key is required when you register an OAuth2 client application in SAP SuccessFactors. The private key must be kept secure under all circumstances. Do not share the private key with others. If you lose the private key, you must create a new certificate.

Example of a public key:

```
-----BEGIN CERTIFICATE-----
MIIB9jCCAV+gAwIBAgIUKR82LgtkNBccdypYD26K87zZ+vYwDQYJKoZIhvcNAQEE
BQAwDTELMAkGA1UECwwCRVAwHhcNMTkwOTI2MDIwNDUyWhcNMTkxMDI2MDIwNDUy
WjANMQswCQYDVQQLDAJFUDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAwKva
NZCOGcuY90/BudS+qQic+A31uM8mLtmI60R1iEjgEWGBCxSiDb2h8mQJiXwkul9W
ebaazP7hkqkdNoJgV/6NE7++GKyyS8fIhJgeWSb6ElMFhjQ0nZKzbZX5ms3I91n
twzkcHtKCQi/gi/Rouhlk/P/QVcrzSgHUHqJNy0CAwEAAaNTMFEwHQYDVR0OBBYE
FHHbgqnnhm3GAJ4gy2IuEDxpLye7MB8GA1UdIwQYMBaAFHHbgqnnhm3GAJ4gy2Iu
```

```
EDxpLye7MA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEEBQADgYEAG5CoqcEy
15vUpj5VfJeR/DS70tPIinp/TCC9kRO/++TSnPbqVcfPr8vIyc4L3MPKjXFBsefE
vtfHGGucVtv5N1+4U/b9NxNFbuH2MP7W3swZ4WM72Na+W6iOhwesOr0p3IcOfxc3
RNCnagFmtbDFxAlPXQ0d+m+N5gxLRoCX1hE=
-----END CERTIFICATE-----
```

Example of a private key:

```
-----BEGIN PRIVATE KEY-----
MIICdQIBADANBgkqhkiG9w0BAQEFAASCAl8wggJbAgEAAoGBAMCr2jWQjhnLmPdP
wbnUvqkInPgN9bjPJi7ZiOtEdYhI4BFhgQsUog29ofJkCYl8JLpfVnm2msz+4ZKp
HTaCYFf+jRO/vhisskvHyISYHlkm+hHpTBYY0NJ2Ss22V+ZrNyPdZ7cM5HB7SgkI
v4Iv0aLoZZPz/0FXK80oB1B6iTctAgMBAAECgYAid5vVsUJ6gt2egHobkF97Rbsu
9PBcW1JtVyUTUW/lLYRIF7VKEirbYm0yO4spOTgozx1dMLmIqqAX6ID9W114kN/g
lz1c2/jMg+YGp+FNCjULygjfIwtGfpX8G0qYWza5oarzVbbGA1cvpHjyNMGV7ure
7syrjIXUighkaKrxgQJBAObVbGTVr/5xxScB1mPYoBe02JMyTzuVW0ts7NyfxXJu
w9vUoMDLV+2wuDE4w8/gUkKf26eojn3kwD708V6lG4kCQQDVrVC7HcXYfU4wkr5S
JPMQzAln0RUf6LgFpgIDPDKpq7VUti1A9aQUbddxlcudFjO57ksr2yU9sOLQgh3A
+2GFAkAWkRDavsVI48h5asWR11C3YJe3tDhow848DncNjpUX/dop+JyKnJaJBzjK
nxkNjomcN9KajnD3v9BHl1ytewi5AkA8IAWscUc/kJrUziXhpWYD3vXykYG5Ndm6
NSkx0dmLprZifNSlB7nAyduqqXTe4eVyNxxN3d9PyZs5ArPuno2lAkAQ8WiHbqGA
JlO6R9+D6HiWywpCaQ0oh6H/+84mb1ew2SUw1mFxROxgfsRVNUe+ahs3nSIhoba0
cqS0ZSBtNDxV
-----END PRIVATE KEY-----
```

3.1.2 Creating an X.509 Certificate in SAP SuccessFactors

You can create an X.509 certificate in SAP SuccessFactors HXM Suite if you're unable to create a self-signed certificate.

Context

> ⚠ Caution
>
> We don't recommend creating the X-509 certificate in API Center and downloading the private key. This method is less secure compared with a self-signed certificate because downloading the private key increases the risk of exposing it. This method should only be used if the client is unable to create a self-signed X.509 certificate.

Procedure

1. Log into your instance as an administrator.
2. Go to *Admin Center* > *API Center* > *OAuth Configuration for OData* and choose *Register Client Application*. You can also access the tool by searching *Manage OAuth2 Client Applications* in Action Search.
3. On the new OAuth client registration screen, choose *Generate X.509 Certificate* and enter the following information:

Option	Description
Issued By	Value set to **`SuccessFactors`**
Common Name	The hostname or IP address for which the certificate is valid. The common name (CN) represents the hostname of your application. It's technically represented by the commonName field in the X.509 certificate. The common name doesn't include any protocol, port number, or path. For example: **`www.bestrun.com`**
Organization	(Optional) The entity to which the certificate is issued.
Organization Unit	(Optional) The organization unit of the entity to which the certificate is issued.
Locality	(Optional) Name of locality of the entity to which the certificate is issued.
State/Province	(Optional) Name of state or province of the entity to which the certificate is issued.
Country	(Optional) Enter a two-letter country code of the entity to which the certificate is issued. A country code represents a country or a region. Example: **`AU`**
Validity	(Optional) The number of days for which you want the X.509 certificate to be valid. If left empty, the validity defaults to 365 days. **i Note** Validity check works only when the *Enable validity check* option is selected.
Enable validity check	Indicates whether or not the system checks the validity of the certificate. If disabled, the certificate never expires. If checked, you can either specify the validity period in days in the *Validity* field, or leave it empty so that the validity defaults to 365 days.

4. Choose *Generate*.

Results

A new X.509 certificate is generated and filled in the *X.509 Certificate* field on the new OAuth2 client registration screen. Continue your registration in Registering Your OAuth2 Client Application [page 14] with this certificate.

> **⚠ Caution**
>
> Both the public key and private key are available to you in the generated certificate. You must save the private key before you register your client application. Only the public key is available for viewing when the client application is registered. The private key must be kept secure under all circumstances. Do not share the private key with others. If you lose the private key, you create a new one.

3.2 Generating a SAML Assertion

Generate a Security Assertion Markup Language (SAML) assertion for requesting an OAuth token. This topic explains how to generate a SAML assertion using the offline tool provided by SAP SuccessFactors.

Prerequisites

You've registered your application in *Manage OAuth2 Client Applications* and obtained the API key for the application.

If you want to use the offline tool to generate a SAML assertion, you need to install Apache Maven in your local environment. Apache Maven is required to run the commands to generate SAML assertions in this task. For more information, see Installing Apache Maven .

Context

You have the following options to generate a SAML assertion:

- (Recommended) Use a third-party IdP that you trust. Refer to the documentation of that IdP for detailed instructions.

  > **i Note**
  >
  > Both SHA-256 and SHA-1 signing algorithms are supported. However, we recommend that you use SHA-256 for better security.

- Use the offline tool.
- Use SAP SuccessFactors IdP. Note that the SAML assertion is valid only for 10 minutes.

  > **⚠ Caution**
  >
  > Do not use this option to generate SAML assertion. It requires you to pass the private key through an API call and is not secure.

SAP SuccessFactors provides an offline tool to generate a SAML assertion for your registered application to access APIs. The SAML generator tool processes the input information offline and generates a SAML assertion without having to expose your private key to the Internet.

Required Elements for Third-Party SAML Assertions

If you choose to use a third-party IdP to generate a SAML assertion, make sure that you follow the SAML 2.0 standard and the following elements are included in the assertion:

> **→ Tip**
>
> SAML assertions are Base64-encoded. To view the detailed information in XML format, decode the assertion using a Base64 decode tool.

Required Elements for SAML Assertions Generated by Third-Party IdP

Element	Description	Example
`<saml2:Issuer>`	Issuer information of the SAML assertion	`<saml2:Issuer>www.myidp.com</saml2:Issuer>`

`<saml2:Subject>`, `<saml2:NameID>`, and `Recipient`	Enter the SAP SuccessFactors user ID that you use to access the APIs in the NameID element. The recipient attribute must be set as the URL of the API server from which you request the OAuth token.	`<saml2:Subject>`
		`  <saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">admin</saml2:NameID>`
		
		`<saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">`
		
		`<saml2:SubjectConfirmationData NotOnOrAfter="2020-08-21T09:23:24.511Z" Recipient="http://<api-server>/oauth/token"/>`
		`  </saml2:SubjectConfirmation>`
		`</saml2:Subject>`

`<saml2:AttributeStatement>` and `<saml2:Attribute>`	The AttributeStatement element must contain the API key (clientId) that you obtained after you register the client application in Registering Your OAuth2 Client Application [page 14].	`<saml2:AttributeStatement>`
		`  <saml2:Attribute Name="api_key">`
		
		`<saml2:AttributeValue xsi:type="xs:string">NDU0MDE0MDkwYj***5YTE5MWIxMTNkNjc1Zg</saml2:AttributeValue>`
		`  </saml2:Attribute>`
		`</saml2:AttributeStatement>`

`<saml2:Conditions>`, `NotBefore`, `NotOnOrAfter`, and `<saml2:Audience>`	The `NotBefore` and `NotOnOrAfter` attributes in the `<saml2:Conditions>` element defines the validity period of the SAML assertion. The `<saml2:Audience>` element is used to tag the SAML assertion. Any value is accepted except empty value. For example, **`www.successfactors.com`**.	`<saml2:Conditions NotBefore="2020-08-21T09:03:24.511Z" NotOnOrAfter="2020-08-21T09:23:24.511Z">`
		
		`<saml2:AudienceRestriction>`
		
		`<saml2:Audience>www.successfactors.com</saml2:Audience>`
		`  </saml2:AudienceRestriction>`
		`</saml2:Conditions>`

Procedure

1. Download the SAML generator tool from 3031657 and extract the files to your local directory.
2. Go to the `SAMLAssertionGen` directory where the extracted files are located, open the SAMLAssertion.properties file with a text editor, and enter the following values:

tokenUrl	Required	The recipient URL from where the access token is to be generated. This is the OAuth API endpoint of your SAP SuccessFactors API server. Example:`https://<api-server>/oauth/token`
clientId	Required	The API key you obtained after you register the client application.
userName	Required	The SAP SuccessFactors username used to access the APIs.
privateKey	Required	The X.509 private key used to sign SAML assertion. It can be the private key of a self-signed X.509 certificate or the private key of an X.509 certificate generated by SAP SuccessFactors.
expireInDays	Optional	Set the expiration days of the SAML assertion. The default value is set to 10000. **i Note** The SAML assertion expiration time doesn't affect the OAuth token expiration time.

3. Save the changes.
4. You have two options to generate a SAML assertion:
   - Option 1: Open a command-line tool and go to the `SAMLAssertionGen` directory. Run the following command:

     ```
     mvn compile exec:java -Dexec.args="SAMLAssertion.properties"
     ```

     This command downloads required files and generates the SAML assertion using the parameters in the properties file. It can take a few minutes to finish.
   - Option 2: If you want to generate a SAML assertion on a machine without Internet access, choose this option.
     1. Open a command-line tool and go to the `SAMLAssertionGen` directory. Run the following command:

        ```
        mvn clean compile package
        ```

> **i Note**
>
> This step must be done on a machine with Internet access. The command generates an executable `SAMLAssertionGen-1.0.0.jar` file in a `/target` directory.

2. Copy the `SAMLAssertionGen-1.0.0.jar` file and the `SAMLAssertion.properties` file to the same folder.

> **i Note**
>
> You can copy the files to a machine without Internet access.

3. Run the following command to generate a SAML assertion:

```
java -jar SAMLAssertionGen-1.0.0.jar "SAMLAssertion.properties"
```

Results

A SAML assertion is generated and displayed in the command-line tool. Here's what the result looks like:

```
The generated Signed SAML Assertion is:
--------------------------------------------------------------------------------
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz48c2FtbDI6QXNzZXJ0aW9uIHhtbG5
zOnNhbWwyPSJ1cm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6YXNzZXJ0aW9uIiBJRD0iOTU1ZDg0ZT
EtNWYyOC00MjY1LTg4YzctYjNiNTVlMDJjNmU5IiBJc3N1ZUluc3RhbnQ9IjIwMjEtMDItMDRUMDY6N
DY6NDkuNzM3WiIgVmVyc2lvbj0iMi4wIiB4bWxuczp4cz0iaHR0cDovL3d3dy53My5vcmcvMjAwMS9Y
TUxTY2hlbWEiIHhtbG5zOnhzaT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS9YTUxTY2hlbWEtaW5zdGF
uY2UiPjxzYW1sMjpJc3N1ZXI+<--MASKED CONTENT-->d3d3LnN1Y2Nlc3NmYWN0b3JzLmNvbTwvc2
FtbDI6SXNzdWVyPjxkczpTaWduYXR1cmUgeG1sbnM6ZHM9Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvM
DkveG1sZHNpZyMiPjxkc3NhbWwyOkF0dHJpYnV0ZVN0YXRlbWVudD48L3NhbWwyOkFzc2VydGlvbj4=
```

Next Steps

Copy the SAML assertion and store it securely in your local drive.

Task overview: Authentication Using OAuth 2.0 [page 13]

Related Information

Registering Your OAuth2 Client Application [page 14]
Requesting an Access Token [page 25]
Viewing the Validity of an Access Token [page 26]

3.3 Requesting an Access Token

With a SAML assertion, you can now call API `oauth/token` to request an access token for authentication with the API server.

Prerequisites

The `/oauth/token` API follows IP restriction settings in the following tools:

- In *Admin Center* > *Password & Login Policy Settings* > *Set API login exceptions...*, you can set access restriction for individual users by IP.
- In *Admin Center* > *IP Restriction Management*, you can set access restriction by IP on the instance level.

Before you request an OAuth token, check the above settings and make sure that the client IP address is allowed to access the corresponding API server. For more information about IP restrictions, see the related information section.

> **i** Note
>
> If both instance-level and user-level IP restrictions are set, a user can access these APIs if either condition is met.

Context

The API returns the token type, expiration time in seconds, and the token value you can use to authorize API requests. An access token expires in 24 hours after it's generated.

Example

Here is a sample request:

HTTP Method	`POST`
URI	`https://<API-Server>/oauth/token`
Headers	`Content-Type: application/x-www-form-urlencoded`

Parameters	• `company_id`: Required. Your company ID. • `client_id`: Required. API key generated in Registering Your OAuth2 Client Application [page 14]. • `grant_type`: Required. Set the value to "urn:ietf:params:oauth:grant-type:saml2-bearer". • `assertion`: Required. Enter the Base64-encoded assertion obtained from Generating a SAML Assertion [page 21]. • `new_token`: Optional. If you have already requested an access token with the same SAML assertion and the token hasn't expired yet, your request returns the same token by default with the remaining time indicated in the `expire_in` field. You can use parameter `new_token=true` to force the server to generate a new access token valid for 24 hours.

Sample response:

```
{
    "access_token": "eyJ0b2tlbkNvbnRlbnQ***ZMm5Tdz0ifQ==",
    "token_type": "Bearer",
    "expires_in": 86399
}
```

Task overview: Authentication Using OAuth 2.0 [page 13]

Related Information

Registering Your OAuth2 Client Application [page 14]
Generating a SAML Assertion [page 21]
Viewing the Validity of an Access Token [page 26]
IP Restrictions

3.4 Viewing the Validity of an Access Token

Use API `/oauth/validate` to verify if an access token is valid.

Prerequisites

The `/oauth/validate` API follows IP restriction settings in the following tools:

- In *Admin Center* > *Password & Login Policy Settings* > *Set API login exceptions...*, you can set access restriction for individual users by IP.
- In *Admin Center* > *IP Restriction Management*, you can set access restriction by IP on the instance level.

Before you use the API, check the above settings and make sure that the client IP address is allowed to access the corresponding API server. For more information about IP restrictions, see the related information section.

> **i Note**
>
> If both instance-level and user-level IP restrictions are set, a user can access these APIs if either condition is met.

Context

An access token expires within 24 hours after it's generated. You can use this API to check if an access token is still valid.

Example

The following sample request shows how to validate an access token:

HTTP Method	`GET`
URI	`https://<API-Server>/oauth/validate`
Headers	`Authorization: Bearer <Your access token>`

A valid token returns status `200 OK` in the header. The response body contains the access token, token type, and expiry time in seconds. Example:

```
{
    "access_token": "<Your Bearer token>",
    "token_type": "Bearer",
    "expires_in": 86312
}
```

Task overview: Authentication Using OAuth 2.0 [page 13]

Related Information

Registering Your OAuth2 Client Application [page 14]
Generating a SAML Assertion [page 21]
Requesting an Access Token [page 25]
IP Restrictions

4 OData V4 Metadata

Learn how to retrieve metadata of an OData V4 service.

An OData metadata document is a representation of the data model that describes the data and operations exposed by an OData service. OData V4 exposes its metadata on the service group level. You can append `/$metadata` to the service root URL and issue a `GET` request to fetch the metadata. For example:

```
GET /odatav4/talent/cdp/Learning.svc/v1/$metadata
```

Metadata Annotations

An OData service can have metadata annotations that define and expose additional descriptive data about the resources and their elements, for example, read and write capabilities, field control metadata, documentation, etc. The `/$metadata` query doesn't include annotations in the response by default. However, you can get the annotations in the following ways:

Request the metadata of an OData service with inline annotations by adding a prefer header `prefer: include-annotations="*"`. For example:

```
GET /odatav4/talent/cdp/Learning.svc/v1/$metadata HTTP/1.1
prefer: include-annotations="*"
```

Request the annotations of an OData service as a separate document:

```
GET /odatav4/talent/cdp/Learning.svc/v1/Annotations HTTP/1.1
```

Refreshing Metadata

In OData V4, you can refresh the metadata of an individual service. The `refreshMetadata` action is available for each service by default. For example, to refresh the metadata of the Learning service:

```
POST /odatav4/talent/cdp/Learning.svc/v1/refreshMetadata
```

A successful operation returns a `204 No Content` HTTP status code.

4.1 Basic Concepts

Learn about the basic concepts in OData v4 in SAP SuccessFactors HXM Suite.

Before you start using the services, it's important to understand the following basic concepts in OData v4.

Concept	Description

Service	A service is a collection of resources exposed by OData including entity sets, complex types, functions, and actions. An OData service is identified by its service root. Below is a pattern of an OData v4 service root:
	`https://<api-server>/odatav4/<accessing-mode>/<namespace>/<service>/<version>/`
	Example:
	`https://<api-server>/odatav4/cal/CalSession.svc/`
	You can further query the resources in the service by appending the resource name:
	`https://<api-server>/odatav4/cal/CalSession.svc/CalibrationSession`

Entity Type, Entity Set, and Entities	An entity type is a collection of entities that share the similar attributes. An entity set is a set of entities of the same entity type. An entity is a real-world instance of the entity type.
Complex Type	A complex type is a list of properties without a key. Complex types are commonly used as property values in an entity or as parameters to operations.
Actions and Functions	Actions and functions are operations exposed by OData that executes custom logic on parts of a data model. Functions are operations that do not allow side effects and must return results. Actions are operations that allow side effects and may return results. A side effect means that the data base state is changed by the operation.
Bound and unbound actions and functions	Actions and functions can be bound to an entity instance or a collection of entities. Unbounded actions and functiosn are static operations.

Related Information

OData Version 4.0 Specifications

4.2 Primitive Types

A list of primitive types supported in SAP SuccessFactors HXM Suite OData v4.

Supported primitive types in OData v4

Type	Meaning
`Edm.Binary`	Binary data
`Edm.Boolean`	Binary-valued logic

Type	Meaning
`Edm.Byte`	Unsigned 8-bit integer
`Edm.Date`	Date without a time-zone offset
`Edm.DateTimeOffset`	Date and time with a time-zone offset, no leap seconds
`Edm.Decimal`	Numeric values with decimal representation
`Edm.Double`	IEEE 754 binary64 floating-point number (15-17 decimal digits)
`Edm.Guid`	16-byte (128-bit) unique identifier
`Edm.Int16`	Signed 16-bit integer
`Edm.Int32`	Signed 32-bit integer
`Edm.Int64`	Signed 64-bit integer
`Edm.Single`	IEEE 754 binary32 floating-point number (6-9 decimal digits)
`Edm.String`	Sequence of characters
`Edm.TimeOfDay`	Clock time 00:00-23:59:59.999999999999

Data type `Edm.DateType` from OData v2 is no longer supported.

5 Supported Features

A list of OData v4 features supported in SAP SuccessFactors HXM Suite.

The SAP SuccessFactors HXM Suite OData v4 API framework is built based on the OASIS Standardized Open Data (OData) Protocol Version 4.01. This topic lists the features from the standard protocol currently supported by SAP SuccessFactors HXM Suite.

For detailed information about each feature, refer to the official OASIS documentation .

Headers

Common Headers

All common headers are supported.

For more information about common headers, see Common Headers .

Request Headers

Supported Request Headers in OData v4

Header	More Information
`Accept`	
`Accept-Charset`	
`Accept-Language`	
`If-None-Match`	
`Prefer`	The following preference values are supported: • `continue-on-error` • `include-annotations` • `maxpagesize` • `omit-values` • `return=representation` and `return=minimal`

For more information about request headers, see Request Headers .

Response Headers

All standard response headers are supported.

For more information about response headers, see Response Headers .

Common Response Status Codes

For more information about response status codes and error codes, see Common Responses [page 37].

Data Request and Modification

Supported Operations

OData v4 supports operations to request and modify data including:

- Query (`GET`)
- Create (`POST`)
- Update (`PATCH` for merging data or `PUT` for replacing data)
- Delete (`DELETE`)

> **i Note**
>
> The upsert operation is no longer supported in OData v4.

System Query Options

The following system query options are supported:

- `$format`
- `$count`
- `$orderby`
- `$top`
- `$skip`
- `$expand`

Built-in Query Functions

Supported Built-In Query Functions in OData v4

Function	Description	Example
`contains`	Returns true if the string contains the substring	`contains(lastName, 'Wilson')`
`startswith`	Returns true if the string starts with the substring	`startswith(username, 'cgr')`
`endswith`	Returns true if the string ends with the substring	`endswith(email, 'sap.com')`
`year`	Returns the year from the date/time value	`year(startDate) eq 2015`
`month`	Returns the month from the date/time value	`month(startDate) eq 9`
`day`	Returns the day from the date/time value	`day(endDate) le 10`

Function	Description	Example
`hour`	Returns the hour from the date/time value	`hour(endDateTime) eq 12`
`minute`	Returns the minute from the date/time value	`minute(endDateTime) eq 0`
`second`	Returns the second from the date/time value	`second(endDataTime) eq 0`
`length`	Returns the length of a string	`length(username) gt 8`
`indexof`	Returns the index of the beginning of the substring	`indexof(username, 'grant') eq 1`
`substring`	Returns the string beginning at the index	`substring(username, 1) eq 'grant'`
`tolower`	Converts the string to lower case	`tolower(firstName) eq 'david'`
`toupper`	Converts the string to upper case	`toupper(firstName) eq 'DAVID'`
`trim`	Trims trailing and leading spaces	`trim(username) eq 'cgrant'`
`concat`	Concatenates two strings	`concat(firstName, lastName)`
`round`	Round to the nearest integer	`round(score) eq 5`
`floor`	Round down to the nearest integer	`floor(score) eq 5`
`ceiling`	Round up to the nearest integer	`ceiling(score) eq 10`

For more information, see Built-in Query Functions .

5.1 Examples

A list of example operations in OData V4.

Query Operations

This section lists only examples that are different from OData V2.

Query with $filter

```
GET /Employees?$filter=reports/any(d:d/age lt 30)
```

Query with $expand and $select/$filter expanded records

```
GET /Employees?$expand=reports($select=name;$filter=age lt 30)
```

Bound Functions and Actions

In OData V4, functions and actions can be bound to an entity instance or an entity collection. To invoke a bound function or action, you must specify an entity or entity collection in the request URI.

Invoking bound functions

```
GET /Category(1)/ProductsByColor(color='red')
```

```
GET /Categories/getCategoriesByType(type='shoe')
```

Invoking bound actions

```
POST /LeaveRequest(4)/Reject
```

```
POST /LeaveRequests/Reject
```

Related Information

https://www.odata.org/getting-started/basic-tutorial/

6 Service Limits

Learn about the framework-level service limits in OData v4.

To ensure optimal availability and performance for customers, we've introduced the following service limits to SAP SuccessFactors HXM Suite OData APIs on the framework level. A module can implement specific limits for their services that override the framework-level limit. In such cases, the module-specific limits apply. Refer to the individual API reference for more information about module-specific service limits.

Operation	Limit
URL length	8,000 characters
Query page size	200
Expand page size	200
Expand depth	2
Total expand size	3 entities
Filter depth	1
Number of filter parameters	20
Length of filter parameter values	100 bytes
Number of literals in a filter clause	200
Total batch request size	50
Inline insert depth	3

7 Timeouts

Understand the timeout scenarios during API calls and learn how to handle timeouts.

Timeout is a common error during API calls when a client waits for too long for the server to response. Timeouts can happen due to various reasons. In this topic, you'll learn the common timeout scenarios and how to handle timeouts.

Type	Timeout	Timeout Error Response (HTTP Code)	Solution
Session timeout	10 minutes (API server) 30 minutes (CF server)	502	Retry is possible. Avoid complex API calls with large data requests.
Gateway timeout	7 minutes	504	Retry gap is 5 minutes. Avoid complex API calls with large data requests.
Client timeout	Client-specific	Client-specific	Timeouts that occurs on the client side are handled by the client. See the client-specific guidelines on how to avoid them.

8 Common Responses

This section lists the common errors in OData v4 in SAP SuccessFactors HXM Suite.

Common OData v4 Responses

HTTP Code	Status	Error Code
200	OK	Successful
201	Created	Created
202	Accepted	Asynchronous request
204	No Content	Requested resource has null value.
400	Bad Request	Bad input parameter. The error message indicates which one and why.
401	Unauthorized	Authentication failure because of invalid credentials, OAuth tokens, or sessions.
403	Forbidden	Role or permission required, or others
404	Not Found	Resource not found such as Entity.
405	Method Not Allowed	HTTP method is not allowed to resources.
409	Conflict	Conflict for optimistic or pessimistic locks
412	Precondition Failed	Header condition such as `if-matched`
413	Payload Too Large	Payload too large (the request payload is larger than the server is willing or able to process).
429	Too Many Requests	Too many requests in a given amount of time
500	Internal Server Error	Servers are not working as expected.
501	Not implemented	Not implemented
503	Service Unavailable	Service unavailable

Error Response Structure

The error response body of OData v4 consists of two levels:

- The first level is the error response body that contains the general error code, message, and details.

- The second level is the module-specific error details. The details section contains one or more errors. The `target` field may refer to a property or other elements.

```
{
  "error": {
    "code": "BadArgument",
    "message": "Please check your inputs",
    "details": [
      {
        "code": "NameRequired",
        "message": "Name must not be empty",
        "target": "name"
      },
      {
        "code": "PasswordNotMeetPolicy",
        "message": "Password must have one upper case letter".
        "target": "password"
      }
    ]
  }
}
```

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 a 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.

www.sap.com/contactsap

THE BEST RUN SAP