SAP SuccessFactors

PUBLIC

Document Version: 1H 2025 – 2025-04-04

Using SAP SuccessFactors Boomi Connectors

THE BEST RUN SAP

Content

1 Introduction

The SAP SuccessFactors Dell Boomi connector extends the SAP SuccessFactors HCM suite SFAPI to the Boomi platform.

About SFAPI Connectors

> ⚠ Caution
>
> As of August 1, 2018, Partner API, SFAPI adhoc, and SFAPI simple entities have been deprecated except for `CompoundEmployee`. Please use OData for new API development and consider moving your legacy applications and integrations to OData API.

The SFAPI is an entity/object-based web services (SOAP) interface that provides typical create, read, update, and delete operations for specified SAP SuccessFactors object types and forms. This connector provides a user-friendly interface for creating and managing SFAPI connections and operations.

There are three kinds of SFAPI object types:

- **Form-based object types**: These are objects that reference custom form objects. These objects have a suffix in the form of "<object type>$<form id>".

  > ⓘ Note
  >
  > The form id is a unique identifier for the form object that can vary from Company Id to Company Id.

- **Standard object types**: These are basic objects that do not have the form ID suffix. The objects names are consistent across Company IDs.
- **Ad-hoc object types**: These are objects that extend ad-hoc reports to the SFAPI. They only support read operation known as "asynchronous query". There are two types of ad-hoc object types:
  - Object types that create new reports when queried. These object types provide access to all the fields of an ad-hoc report. These object types are all prefixed with `ADHOC_`.
  - Object types that represent saved reports created in the SAP SuccessFactors *Adhoc Report Builder* user interface. These expose only a subset of the fields defined in the corresponding `ADHOC_ report` object type. These object types are named `AdhocReport_<reportId>`.

Asynchronous query operations run "asynchronously": the query is executed in the HCM suite cloud and a "get job status" operation is run periodically, waiting for the report to be queued and completed. Once completed, the results are fetched by the SAP SuccessFactors Connector.

Best Practice: Asynchronous queries are very useful because they provide the ability to join multiple related object types. Synchronous queries only allow return records from a single object type. For example, an asynchronous query from an Adhoc object type can return employee data that includes personal information, job history and postal address information related to each employee.

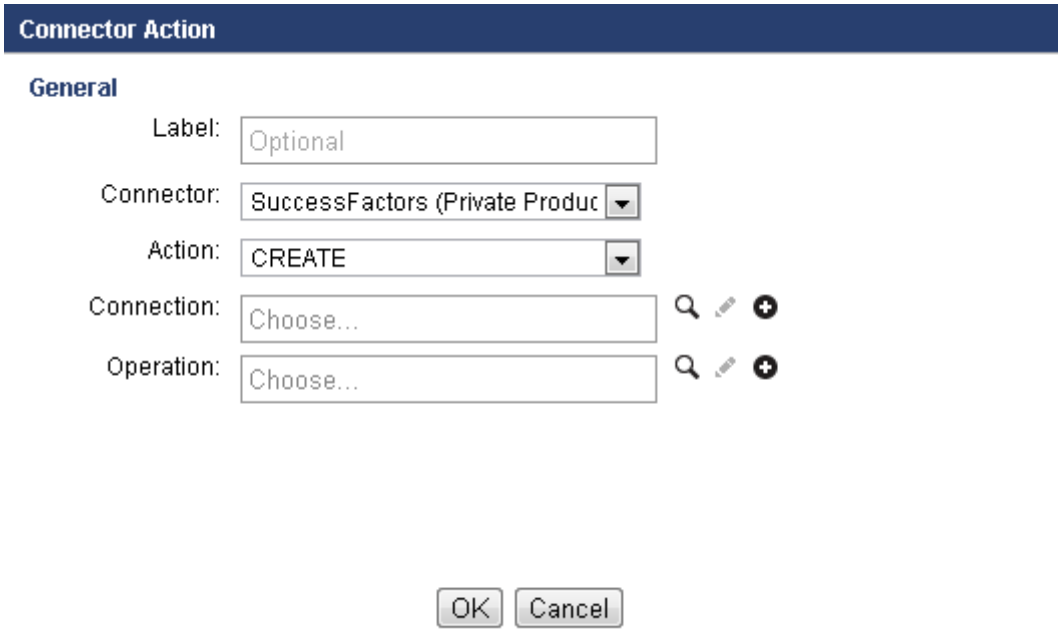
The steps that occur when running an asynchronous query are:

1. The report is created from query and saved as an Adhoc Report (ADHOC_ object types only)
2. A report job is submitted to the scheduler for execution. (Both ADHOC_ and AdhocReport_/saved reports).
3. The connector periodically executes a “get job status” operation waiting for the report to complete execution.
4. The report definition is deleted from the Adhoc Saved Reports (ADHOC_ object types only).
5. The results are downloaded from SAP SuccessFactors to the connector.

Note: Even though the Adhoc SFAPI returns CSV files in either compressed or non-compressed text formats, the Boomi Connector converts these results to XML format, matching the XML profile generated by the connector when an Adhoc object type is imported into the connector.

Best Practice: Report execution results are stored in SAP SuccessFactors for 30 days. You can log into SAP SuccessFactors and download any report execution form the Saved Reports page.

To find the name of the report that is executed, open the Boomi execution log and scroll down to where the report name is logged. Both the report name (prefixed with SFAPI) and the Task/Job ID will appear in the Boomi log file.



Related Information

Deprecation of Partner API, SFAPI Adhoc, and SFAPI for Simple Entities

1.1 Terminology

- "Object Type" is synonymous with "entity". Boomi's preferred term is "object" while the SAP SuccessFactors SFAPI materials refer to "entities".
- Company ID is synonymous with tenant or cloud-instance. Specifically, the company ID represents a specific instance in a data center.

2 Creating an Integration Process

Create a process to start your integration flow.

Procedure

1. On the *Welcome* tab of the *Integration* screen, choose *Create* > *Process*.

   A *New Process* tab displays.
2. On the *Start Shape* panel, complete the following steps:
   a. Choose *Connector* as the *Type*.
   b. On the *General* tab, enter the following information:

Display Name	A unique name for the shape.
Connector	Select *SuccessFactors* from the list.
Action	Choose one of the following actions to perform: • UPSERT • DELETE • UPDATE • CREATE • QUERY • EXECUTE
Connection	Select an existing connection from the list or choose + to create one. For more information, see Creating a Connection [page 7].
Operation	Select an existing connection from the list or choose + to create one. For more information, see Operation [page 10].

   c. On the *Parameters* tab, choose the parameters you want to apply to the process. You can import, add, edit, and remove parameters here.
3. Choose *OK* to save the shape.
4. On the *New Process* tab, you can optionally rename the process, add a description, and change the folder to store the process.
5. Save the process.

2.1 Creating a Connection

Create a connection to an SAP SuccessFactors API endpoint so that you can use it in your Boomi Connector.

Context

The *Connection* tab is where you specify the data center, company ID, and authentication details for performing API transactions. Connecting to an existing company ID requires both the endpoint URL to the data center and the company ID representing an SAP SuccessFactors instance in the data center. Please refer to the Tools and References [page 47] for more information on configuring SFAPI connections.

Procedure

1. On the *Welcome* tab of the *Integration* screen, choose *Create* > *Component* > *Connection*.

   The *Create Component* dialog pops up.
2. Choose *Connection* as the component type, enter a name, select the folder to store the component, select *SAP SuccessFactors* as the connector, and choose Create.
3. On the *Connection* tab of the new component, enter the following information:

Option	Description
Endpoint	Select an endpoint from the dropdown menu. If the URL isn't listed in the dropdown menu, select the Other option in the menu and manually enter the URL in the Other Endpoint field. There's an endpoint and accompanying WSDL for each data center. The endpoint is specific to the global data center where the company instance is deployed. It specifies where to send the SOAP XML request payloads.
Other Endpoint	If the endpoint isn't listed, enter it manually here.
SFAPI endpoint suffix	Override the URL suffix of the SFAPI endpoint (/sfapi/v1/soap). **Note** Don't use the SOAP version 1.2 end point (/sfapi/v1/soap12).
Options	• Enable SFAPI Entity Import: Enables access to the SFAPI.

Option	Description
	• Enable SF OData Entity Import: Enables access to SAP SuccessFactors OData API. OData entities appear on object list as SFOData.[EntityName]
SF OData endpoint suffix	Override the default URL suffix of the SF OData endpoint (/sfapi/odata/v2/).
Company Id	Company ID o your SAP SuccessFactors instance. Required to run SFAPI transactions.
Username	A valid username in the specified instance. The user must have permissions for SFAPI login and read/write operations of the desired data objects. **Note** Prefix this username in the form "sfapi-<vendor or module use>" in lower case, for example: sfapi-ec, sfapi-rcm, sfapi-saperp, sfapi-sapbyd, etc. This information helps administrators identify that this user is being used for integration, and not to disable the user, which would disable SFAPI access and break integration processes.
Password	Enter the password for the username. **Note** HTTP Basic Authentication has been deprecated. We recommend that you use OAuth 2.0 to authenticate API users. For more information, see the Related Information section.
OAuth2 Client Id	The API Key generated in *Manage OAuth Client Applications* after you register your application. For more information about registering OAuth client applications, see the related information section.
X.509 Private Key	Private key of the X.509 certificate that you use to register the client.
OAuth2 SAML Assertion Field	The SAML Assertion Field Name set in process.
OAuth2 SAML Assertion	The SAML assertion you use to authenticate user.
Default Batch Size	Specify the default page size to use for write operations. The value can be from 1 to 800. The default value is 200. Set a reasonably large number to limit the number of individual SOAP transactions, but not so large that slower SOAP transactions exceed 1–2 minutes and risk timing out.

Option	Description
	Note This value is shared for all operations that reference the connection but can be overridden on a per operation basis.
Default Query Page Size	Specify the default page size for query operations. It's similar to *Default Batch Size* but it controls the size of the returned query results. Set a reasonably large number to limit the number of individual SOAP transactions but note the maximum value supported by the SFAPI is 800. **Note** This value is shared for all operations that reference the connection but can be overridden on a per operation basis.
Default Async Query Timeout time (async query only)	Specify the time the system waits for the completion of an asynchronous query job for an ad-hoc report. Set this value large enough to allow for slow running queries and long scheduler queues. **Note** This value is shared for all operations that reference the connection but can be overridden on a per operation basis.
Default Sleep/Wait time (async query only)	Set the frequency that the connector checks the status of an asynchronous query job of an ad-hoc report. A longer sleep/wait time limits the number of "get job status" transactions. Set a relatively large value to limit the number of "get job status" transactions that occur while waiting for the query to complete. **Note** This value is shared for all operations that reference the connection but can be overridden on a per operation basis.
Object ID Type List	Enables the setting of Form type entity IDs from a Connection Environment Extension. Specifies list of form IDs to use in format PMForm=PMForm$1;JobRequisition=JobRequesition$13; In this case, all PMForm type IDs are set to PMForm$1 in order to function for a specific SAP SuccessFactors instance.

4. Save the connection.

Results

The connection is available to choose when you create an integration process.

Related Information

Authentication Using OAuth 2.0
Registering Your OAuth2 Client Application [page 37]

2.2 Operation

The Operation specifies what action to be performed on which object type.

The actions supported are:

- Query (OData and SFAPI) - Read a set of objects optionally sorted and limited to those records that meet a specified search criteria (filter).
- Create (SFAPI only) - Insert a new record. This operation returns the new ID of the inserted object.
- Update(SFAPI only) - Update the specified field values of an existing record. The ID specifies the record to update.
- Delete (SFAPI only)- Remove the record. The ID specifies the record to delete.

  > **Note**
  >
  > Very few SAP SuccessFactors object types support the Delete operation. Most perform "soft deletes" that update the record with a status flag of "Inactive".

- Upsert (SFAPI only)- Write a record specified by a unique External ID. If a record with that external ID exists, the record will be updated. Else a new record will be created.
- Execute - (Odata Only) - This operation is for writing SFOData entities. The type of write operation is specified by the ODATA Method/Operation.

> **Note**
>
> For some object types, an internal ID will be returned as generated by the system.

2.2.1 Creating an Operation with the SuccessFactors Import Wizard

Context

The only way to create a new SuccessFactors Operation is to Import an operation from a specific SAP SuccessFactors instance using the *SuccessFactors Import Wizard*. This is because each instance may have a unique set of entities/forms/object types each with their own unique custom fields. The goal of an import operation is to select an object definition from the instance and generate the input and output XML profiles for the object definition and the respective action/operation type.

For more information on object types and allowed operations for an object, please refer to the *SFAPI Data Dictionary* in the Admin Tools web page for the specific instance.

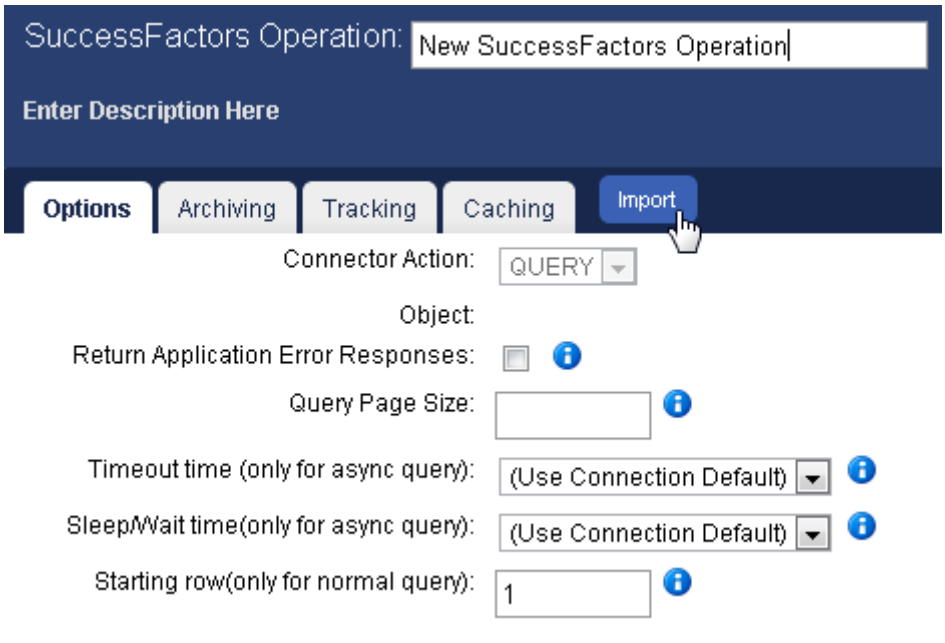
Before importing a new operation you must first choose the Connector Action that will be used. There are two ways to create a new SuccessFactors Operation and specify the Connector Action:

Procedure

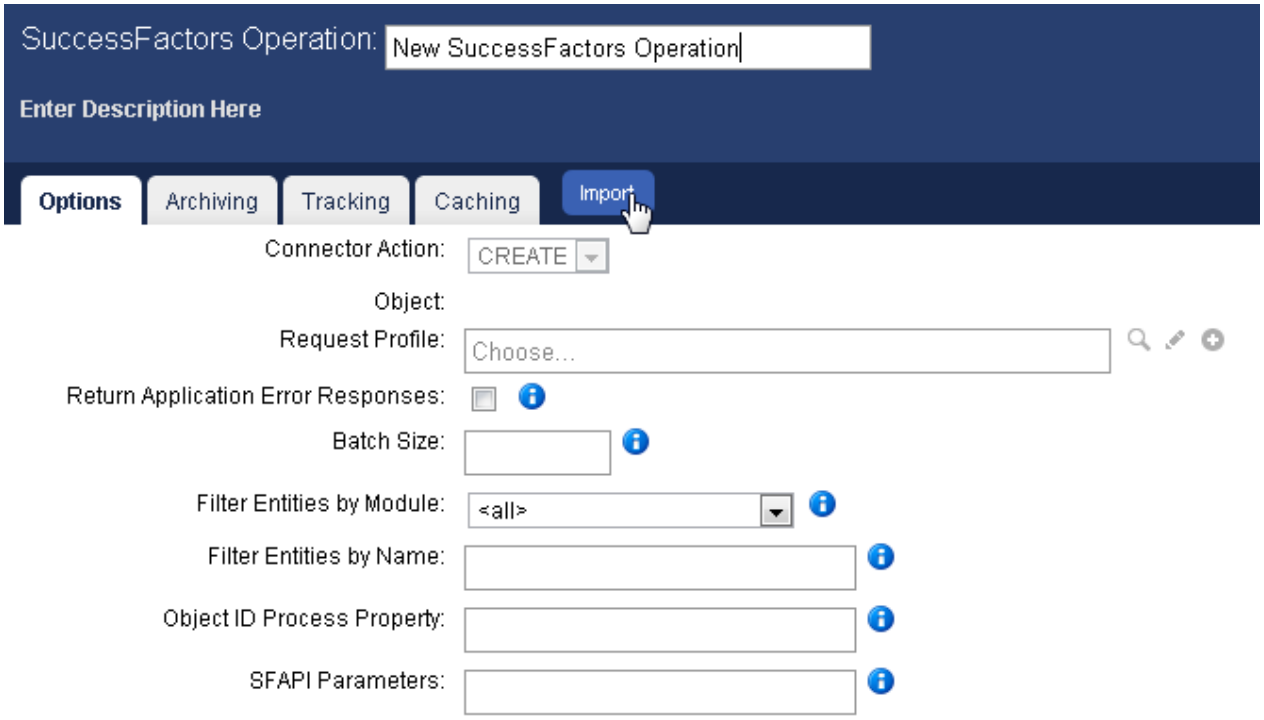
1. Select Create Component from the Component Explorer. In this case you can select the Connector Action from the new operation dialog that appears.
2. Create a new operation from an existing Connector. In this case the Connector Action will be disabled because the operation is specified in the Connector.

In either case, a blank SuccessFactors Operation will be displayed.

> **Note**
>
> The Query Operations and Write (Create/Update/Upsert/Delete) operations have slight differences (see images below).



A New Query Connector



A New Write Operation

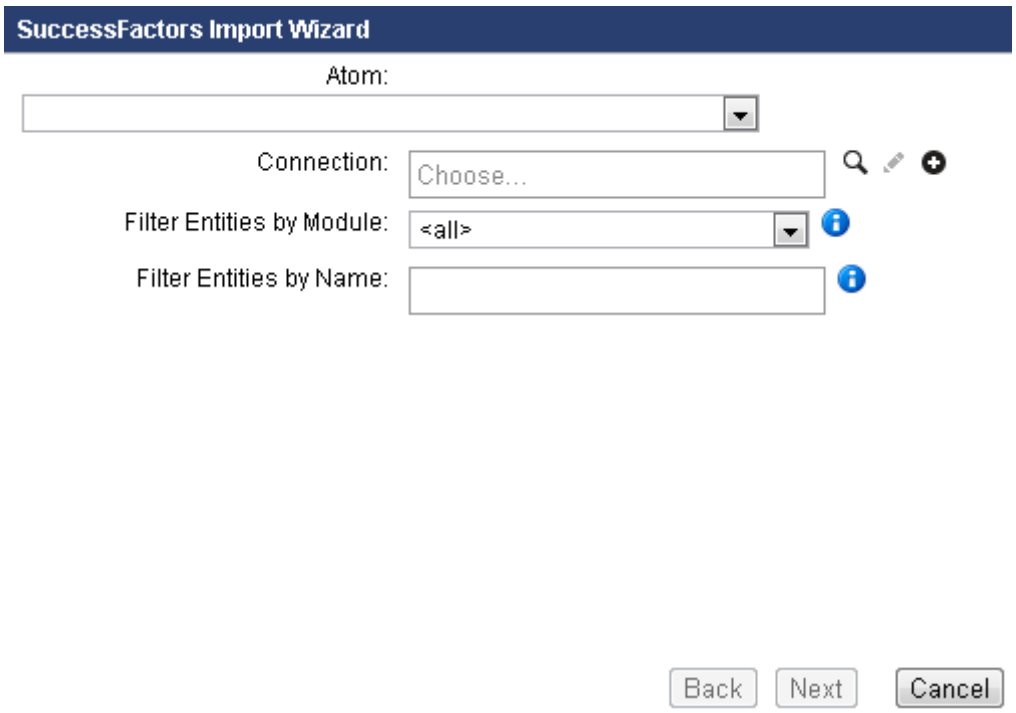
The full object list for a specific Company ID can be viewed using the Admin Tools *SFAPI Data Dictionary*.

3. Launch the SuccessFactors Import Wizard by clicking on the *Import* button.

   The first page of the SuccessFactors Import Wizard appears.

2.2.1.1 Importing Object Definitions for a Company ID

Specify an SAP SuccessFactors instance for importing object definitions.



The first page of the wizard is used to specify from which SAP SuccessFactors instance object definitions will be imported. An instance is represented by its company ID.

2.2.1.1.1 Atom

Importing is an active operation that must execute on a Boomi Atom server.

Best Practice: We recommend that you use separate Atom servers before deploying your Boomi process (design time) and after it's deployed (runtime). Sharing an Atom server between design time and runtime processes can cause invalid session issues.

> **Note**
>
> For security purposes, the Import operation is disabled on Boomi clouds. Hence a standalone atom (or a development cloud if available for the Boomi account) must be used. If necessary, install a standalone atom on a local computer for use when importing.

2.2.1.1.2 Connection

The Connection specifies from which SAP SuccessFactors instance to import object definitions. For more information please refer to Creating a Connection [page 7].

> **Note**
>
> It is important to consider what instance the import is performing against because each instance may have a different set of forms, object types and custom fields.

2.2.1.1.3 Filter Entities by Module

Selecting from this list will filter the list object definitions pertaining to the HCM Suite module selected. To see all object types, select *<All>*. Selecting specific a specific HCM suite module from the list will filter accordingly.

- Employee Central
- Goal Management
- Performance Management
- Platform
- Recruiting

2.2.1.1.4 Filter Entities by Name

Alternatively, you can perform a "wildcard" search for object types by specifying a search pattern in this field.

> **Note**
>
> Text in this field will override any selection made in the *Filter Entities by Module* list. For example, specifying "erp" will return only PerPerson, PerPersonal, PerPhone, UserPhoto and UserPhotoSource.

If this field is left blank, the objects specified by the *Filter Entities by Module* will be returned.

Best Practice: The Boomi Connector framework doesn't allow more than 500 entity types to be listed. If your HCM suite tenant includes more than 500 entity types, you must provide a filter to limit the list to be smaller than 500 records.

When the information is specified, click on *Next* to display the object list.

2.2.1.1.5 Set Min Occurs to 1 for all Fields

All elements will have min occurs set to **1** for write operations. This is useful to clear or set all fields of an object type, in conjunction with the *Respect Min Occurs?* XML profile option.

2.2.1.2 Choosing the Object

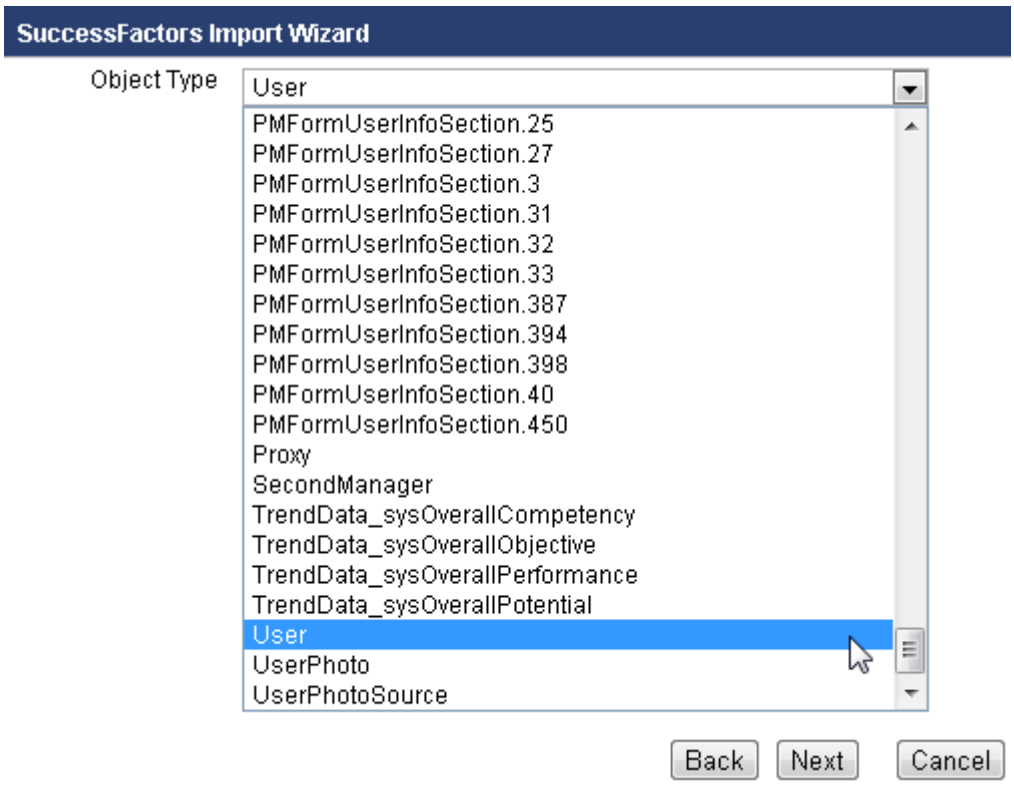
Choose an object to import data in the *SuccessFactors Import Wizard*.

Context

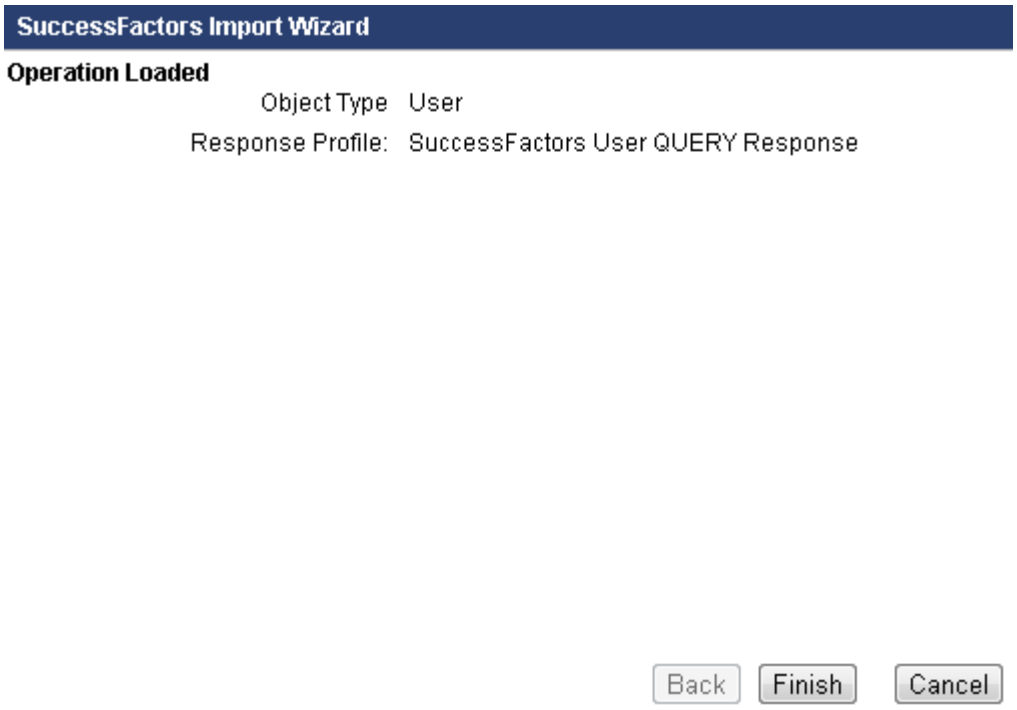
The second page of the *SuccessFactors Import Wizard* displays a list of object types for the company ID specified on the first page.

Procedure

1. Choose an object from the list to create the specified action/operation type for that object type.



After a few moments, the operation will be populated with the object specific information and respective XML profiles will be created.



2. Choose *Finish* to complete the generation of the SuccessFactors Operation.

   Note: If the selected operation/action type is not supported for the selected object type, an error will be displayed at this point. For more information, please refer to the *Legacy SFAPI Data Dictionary* in Admin Center in your SAP SuccessFactors instance.

2.2.1.3 Request and Response XML Profiles

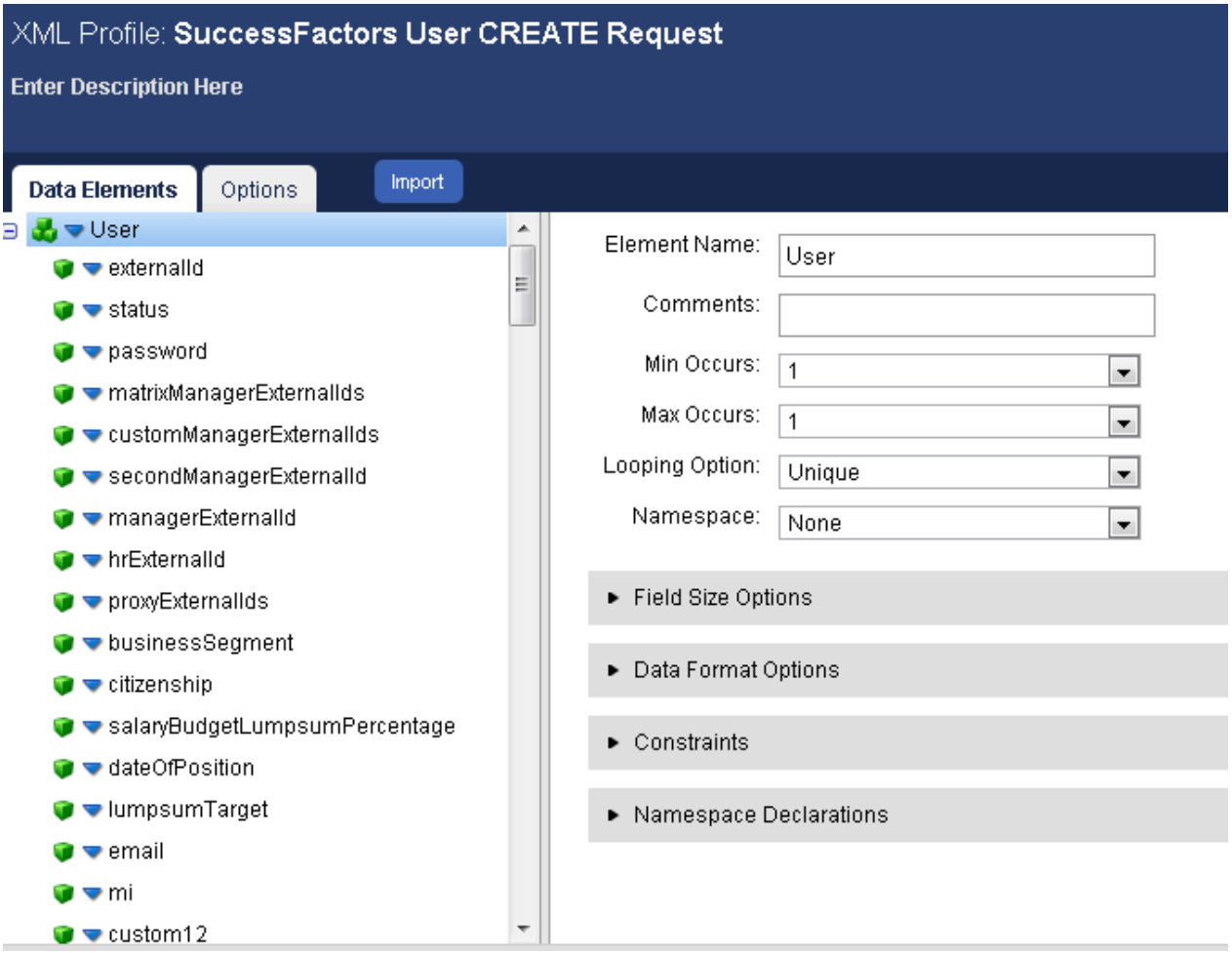
The *SuccessFactors Import Wizard* automatically generates Request and Response XML Profiles.

> ⓘ Note
>
> Field IDs are used as elements in the XML Profile. Field labels, if they exist, are indicated in the Comment field displayed when clicking on a specific field ID element. For saved report object types, (Those with the AdhocReport_ prefix, the Comment field will contain the descriptive name of the saved report.

2.2.1.3.1 XML Profiles for Write Operations

Write Operations [page 31] including Create, Update and Delete operations have both a Request and Response profile.

The Request Profile for a Delete Operations contains only an element to specify the ID for the record to delete. Request profiles for other write operations are those fields indicated as Insertable (Create operations), _Updateable (Update operations) or _Upsertable (Upsert operations) per the object type specification in the *SFAP Data Dictionary*.

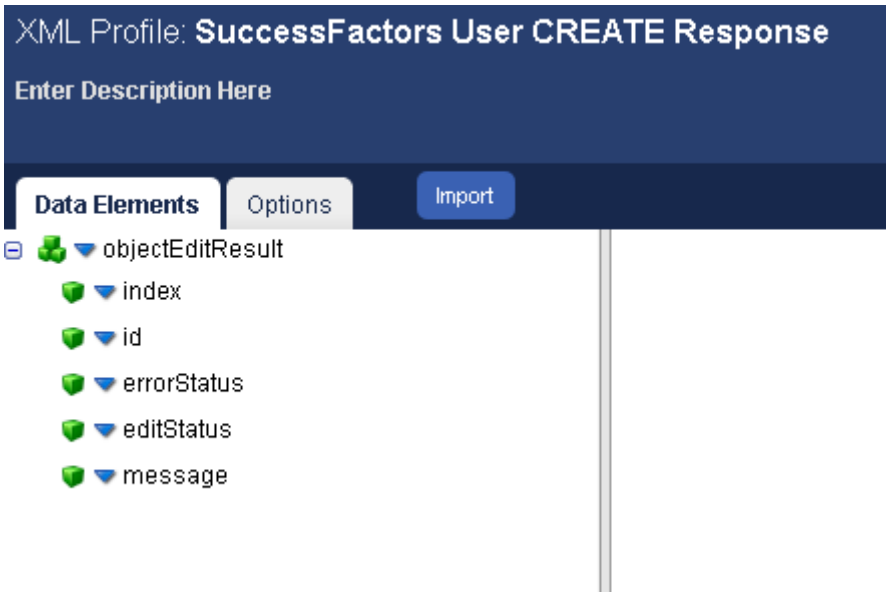


Request documents will have Constraints set according to the SFAPI Data Dictionary metadata for each field. If a maximum length is specified for a field, this value would be reflected in the Constraint section. If the field is required, Min Occurs will be set to **1** and the Minimum Length constraint will be set to **1**.

For update and upsert operations for non-effective dated HCM suite objects, you can leave an existing field value unchanged by leaving the field value blank. To clear a value from an existing record, set the Min Occurs to **1** and and select the *Respect Min Occurs?* option on the *Options* tab.

Upserts of effective data HCM suite objects have a different behavior. By default, leaving a field blank will initiate a NO_OVERWRITE behavior where the field for a new effective dated record will be set to the value for the field in the previous record. Similarly to above, to clear a field and prevent the previous value to be propagated, set the Min Occurs to **1** and and select the *Respect Min Occurs?* option on the *Options* tab.

Write operation Response Profiles provide the status of writing each document and any error indication. An index is also provided to associate specific documents to specific response elements. For more information, please refer to the *SFAPI Error Messages Guide*.

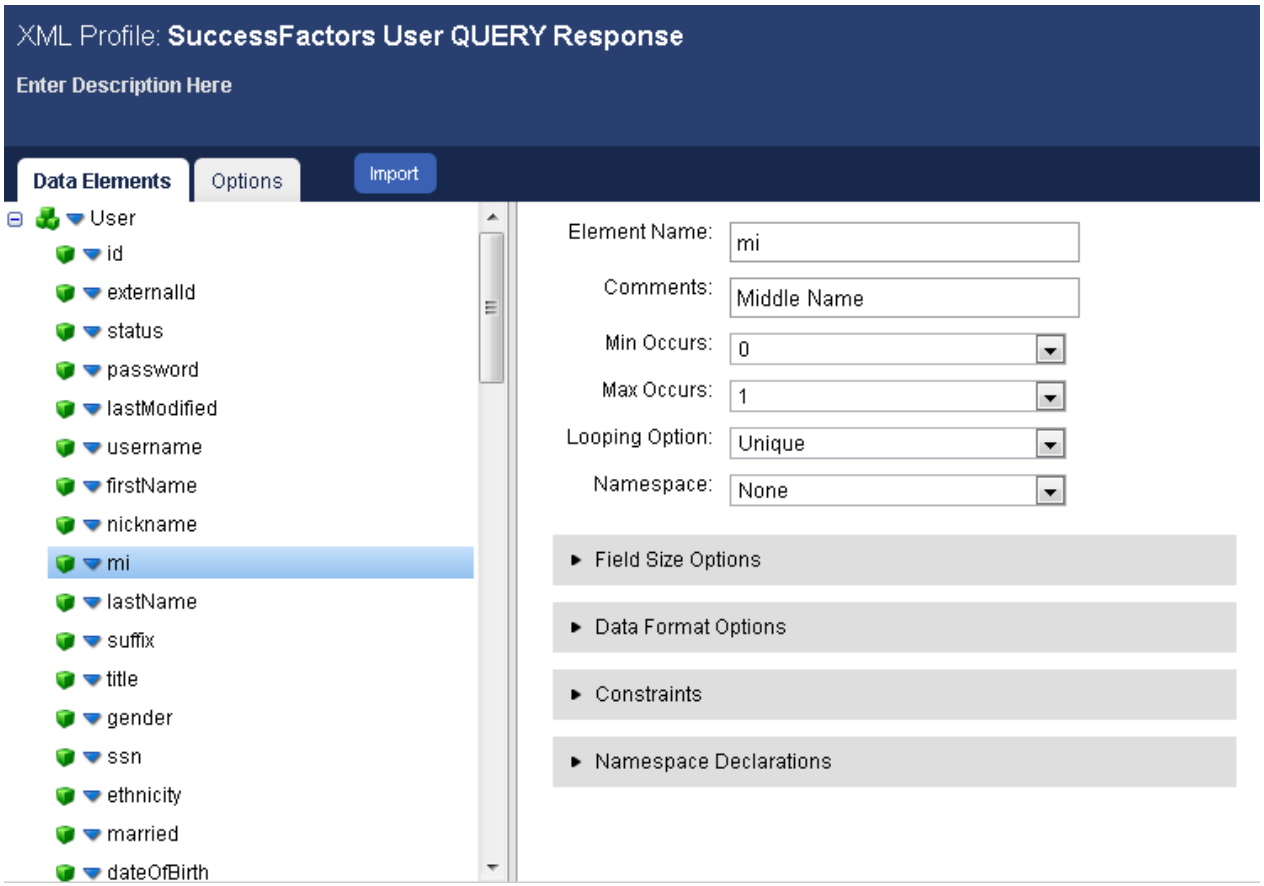


> ⓘ Note
>
> The index is a 1 based ordinal that corresponds with the respective input documents. This ordinal can be used to correlate the input data with each respective edit result.

2.2.1.3.2 XML Profile for Query Operations

Query Operations [page 20] only have a Response Profile.

> ⓘ Note
>
> The only fields that will appear on the Response Profile are those fields indicated as *Selectable* or *Constrainable* per the object type specification in the *SFAP Data Dictionary*.



Adhoc XML Response structure

2.2.1.3.3 Picklist Fields

Picklists fields are associated with SAP SuccessFactors picklist tables. These are name/value pairs that appear in drop down lists within the SAP SuccessFactors user interface pages.

A picklist entry consists of:

1. The ID of the picklist. This ID is the same for each entry that appears in each individual drop down list.
2. The picklist label is displayed in the user interface and on reports. This label will be displayed in the language of the SFAPI user.
3. The picklist option ID. This integer value is automatically generated when importing a picklist to a SAP SuccessFactors company ID.
4. An optional externalCode that provides an alternate value that can be associated with a picklist entry.
5. A status field indicating whether or not the picklist is Active or Intactive (soft deleted).

The values contained in picklist fields are limited to the values contained in the associated picklist.

Enumerations are similar to picklist but are not configurable. Rather the possible set values are hard coded by the object type. The picklist status field mentioned above is an example of an Enumeration field type.

Best Practice: Always use an SFAPI user that is associated with the en_US locale. This will ensure all labels will use the en_US picklist labels.

When a field is associated with a picklist, the picklist ID will be included in the Comment for the field. Similarly if the field is associated with an Enumeration, the set of possible values will be included in the Comment for the field.

Best Practice: It is often useful to translate field values from one picklist field value to another. For example, asynchronous query results will always contain the picklist label value, as will other object types. It may be necessary to translate this value to the externalCode for writing to an external system (for example translate United States to "USA"). It is tempting to use a hard coded Cross Reference Table to perform this but it is a much better approach to use the picklist values because the spelling of the labels is guaranteed to match. If the label spelling fails to match a Cross Reference Table entry, the field will be translated to a blank and may result in data integrity issues.

Many object types require the picklist option ID when writing values to fields. Similarly a translation can be performed to convert an externalCode value to the associated picklist Option ID.

The best way to perform these translations is to use the SAP SuccessFactors Common Component to load picklists into memory. Then use another Common Component map function to perform the translations.

Alternatively, picklists can be queried and written to Document Cache steps, but this will be a very cumbersome approach as it will require a Document Cache for each picklist ID.

2.2.2 Configuring SuccessFactors Operations

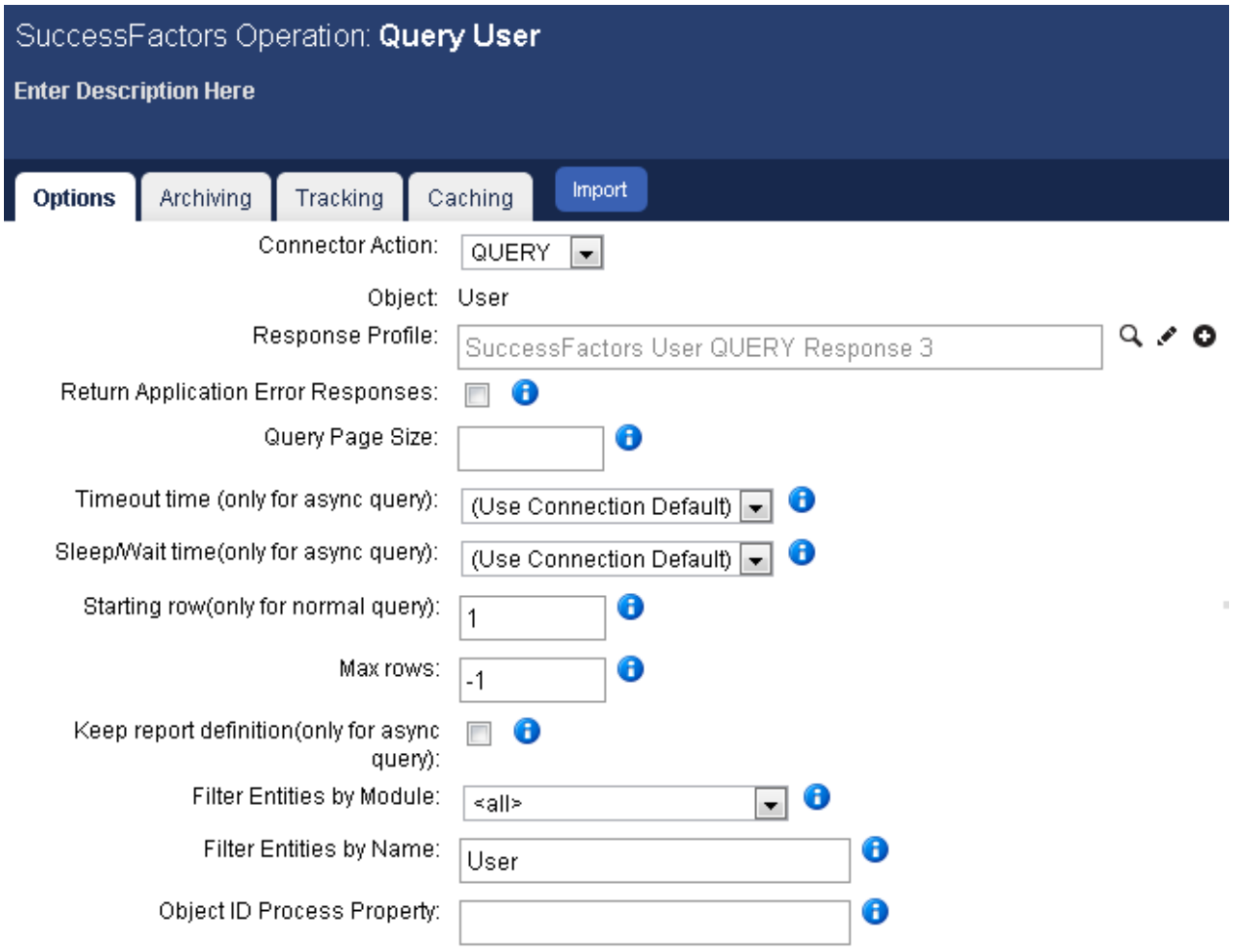
Existing Operations will have the Object type and the Action defined as specified during the Import process. These fields cannot be changed. To define a new Action/Operation type or to generate an operation for a different object type, rerun the *SuccessFactors Import Wizard*.

Best Practice: Specify a small set of Tracked fields on the Tracking tab for ID, external ID and/or other record identifiers such as last name, email, etc. This makes it easier to identify documents when analyzing process executions. Note tracked fields must be defined for the Account on the Setup page before the are available on the operation Tracking tab.

Once the operation definition has been imported, query and write operations can be configured using the SuccessFactor Operation dialog windows.

2.2.2.1 Query Operation

This action queries records in the SAP SuccessFactors object defined in the Create operation. All records will be returned as individual **Boomi Documents**. Paging will be performed automatically according to the `Page Size` parameter.



This action returns zero to many object record documents from a single Query request based on zero or more "filters".

2.2.2.1.1 Connector Action

Indicates the operation is indeed a Query operation.

> **Note**
>
> If you change the Connector Action, a new import will be required and the existing query settings will be deleted.

2.2.2.1.2 Object

Defines the object on which the operation is executing. This is only set during an import operation.

2.2.2.1.3 Response Profile

The XML profile that defines the XML schema/structure of the response document returned by the operation. For more information regarding profiles, please see *XML Profiles*.

2.2.2.1.4 Return Application Error Responses

Defines whether to return exception responses through the process or fail at the connector level.

Best Practice: Leave *Return Application Error Responses* unchecked and catch the errors using a Try Catch step.

2.2.2.1.5 Page Size

Overrides the write Page Size specified in the *SuccessFactors Connection*. For more information please refer to *Connection Page Size*.

> **Note**
>
> If you override this value, you cannot control it from a Process Extension.

2.2.2.1.6 Starting Row

For synchronous queries only. Indicates to skip the first number of records specified. For example, a value of 10 skips the first 9 records. A value of 1 (default) returns all records.

2.2.2.1.7 Max Rows

Specifies the maximum number of records/documents to return. A value of -1 indicates returning all records.

Best Practice: Boomi allows querying a maximum of 10 documents during a *Run as Test*. Change this value to 10 to allow executing a *Run as Test* without error. *If you do this, don't forget to change it back to -1 before deploying the process.

For simple processes, configure the Connector in the Start step which automatically prompts to override a large number of returned records and rerun with only 10 documents.

Best Practice: For unit testing, disconnect the Connector and attach a Message step that contains a test document. This allows your process to Run as Test much faster, especially when executing longer running Asynchronous Queries.

You can even simulate multiple documents by adding a Split Documents Data Process step after the Message step. Configure the Split Documents to split by line with no message header|

2.2.2.1.8 Timeout Time

For Asynchronous Queries Only . Overrides the value specified in the *SuccessFactors Connection*. For more information please refer to *Connection Timeout Time*.

> ⓘ Note
>
> If you override this value, you can not control it from a Process Extension.

2.2.2.1.9 Sleep/Wait Time

For Asynchronous Queries Only. Overrides the value specified in the *SuccessFactors Connection*. For more information please refer to *Connection Sleep/Wait Time*.

> ⓘ Note
>
> If you override this value, you can not control it from a Process Extension.

2.2.2.1.10 Keep Report Definition

For Asynchronous Queries Only. Selecting this option will leave the report definition in the Adhoc Saved Reports. This option only works when directly running queries against report subdomain schemas (Object types with a prefix of "ADHOC_").

> ⓘ Note
>
> This option has no effect when running Asynchronous Queries on "saved reports" (Object types with a prefix of "AdhocReport_").

Best Practice: Saving an Adhoc report is useful if you like to rerun a report directly from the Adhoc Report Builder user interface in SAP SuccessFactors.

2.2.2.1.11 Remove Namespaces

This will remove the namespaces from the response payload of the EmployeeCompound entity so that they will match the generated XML profile. This was added for compatibility with old operations and it is recommend you turn this on for EmployeeCompound queries.

2.2.2.1.12 Results Format(async query only)

Specifies the format of the document returned from the Asynchronous Query Operation.

- XML - returns the results as multiple documents in XML format.
- Split CSV Documents - returns the results as multiple documents in CSV format. Each document has a header. The header names match the element names of the XML format.
- Single CSV Document - returns the results as a single document. The headers match the field labels.
- Compressed (Zip) - Same as Single CSV Document but the document is in Zip format.

2.2.2.1.13 Object ID Process Property

(Optional) Specifies the name of the process property that contains the object type. This overrides the object type ID extracted during Import. This is generally used for Form object type types to allow the object type ID to be specified dynamically as an Environment Extension.

Best Practice: you should always use this field for Form object types because the integer component of a Form object type ID (For example, "PMForm$22") can vary from instance to instance.

2.2.2.1.14 Filter Process Property

Context

(Optional) Specifies the name of the process property that contains the object type. This overrides the object type ID extracted during Import. This is generally used for Form object type types to allow the object type ID to be specified dynamically as an Environment Extension.

To use this feature:

Procedure

1. Create a Set Properties step that sets the value of a Dynamic Process Property to be the SFQL/Odata URI string you wish to use.
2. Enter the name of the the Dynamic Process Property in this field.
3. When you execute the process, the contents of the property are used for the filter

   Best Practice: you must exam the Boomi Log File connector entries to verify that your filter was generated properly.

   OData Example: set the contents of the process property to &fromDate=2013-01-01&toDate=2014-12-31 in order to perform a date range query for an effective dated entity. Similarly you can perform an as-of-date query using &asOfDate=2014-12-31. You can also set the $filter= uri parameter but you must be knowledgeable of the odata filter syntax to do so.

   SFAPI Example: set the contents of the process property to companyTerritoryCode=XXX AND department=YYY in order to override the entire SFQL WHERE clause

2.2.2.1.15 SFAPI Parameters

(optional) For specifying SFAPI SFParameters. For example: traceLevel=detailedLogging.

2.2.2.1.16 OData Import Depth (import only)

Determines how deep to traverse associations/navigation properties to the entity in order to do join/expand query operations. For example, a depth of 2 for the SFOData.PerPerson entity will result in the ability to query/join down to related entities like SFOData.JobInfo. The larger the depth, the deeper and more complex the query will result.

> ⚠ Caution
>
> WARNING: Deep queries will result in all fields selected by default. You should turn off only those fields necessary our your query will not perform well.

2.2.2.1.17 OData Allow Circular References (Import Only)

Allows query results to include self references and circular references to same entity type.

2.2.2.1.18 Filter Entities by Module (Import Only)

For Import Only. For existing operations this field has no effect and will only reflect what value, if any, was used during import.

2.2.2.1.19 Filter Entities by Name (Import Only)

For Import Only. For existing operations this field has no effect and will only reflect what value, if any, was used during import.

2.2.2.1.20 The Query Operation sub tabs

Upon complete of an Import of a query operation, three sub tabs appear in the Query Operation dialog: Query Fields, Query Filter, Query Sort. The following sections describe these sub tabs.

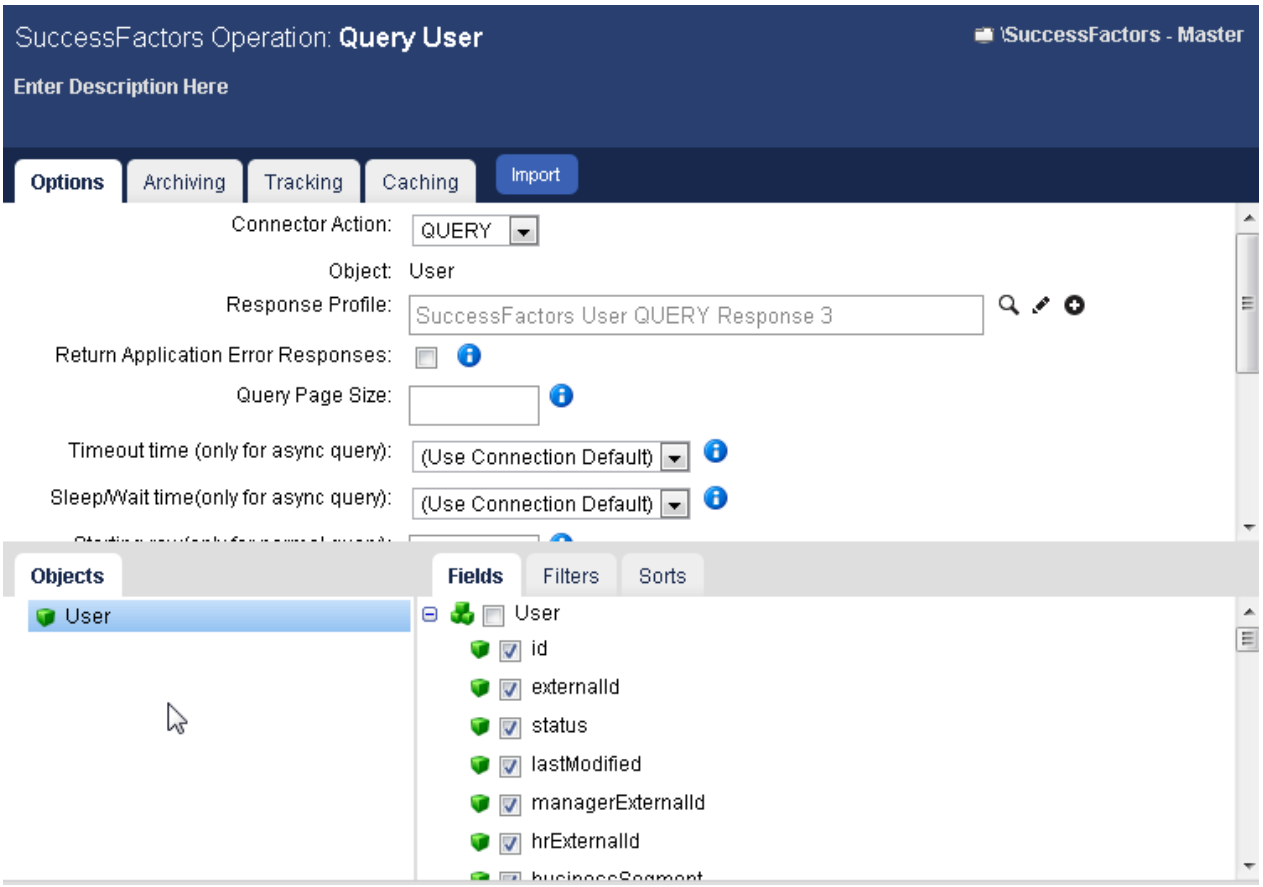
2.2.2.1.21 Query Fields

This sub tab specifies what subset of fields to return in the resultant document. Select the topmost checkbox to select all the fields.

Note: The only fields that will appear on the Query Fields sub tab are those fields indicated as *Selectable* per the object type specification in the *SFAP Data Dictionary*.

Deep OData queries are supported by selecting fields within the field hierarchy. Any field selected at the top level will be included in the OData $select statement. Any fields selected in child entities will cause that entity to be included in an OData $expand operation, causing all the fields in the entity to be joined into the query

Best Practice: By Default, all fields will be selected for query. This can result in extremely complex and poorly performing queries, especially for OData.

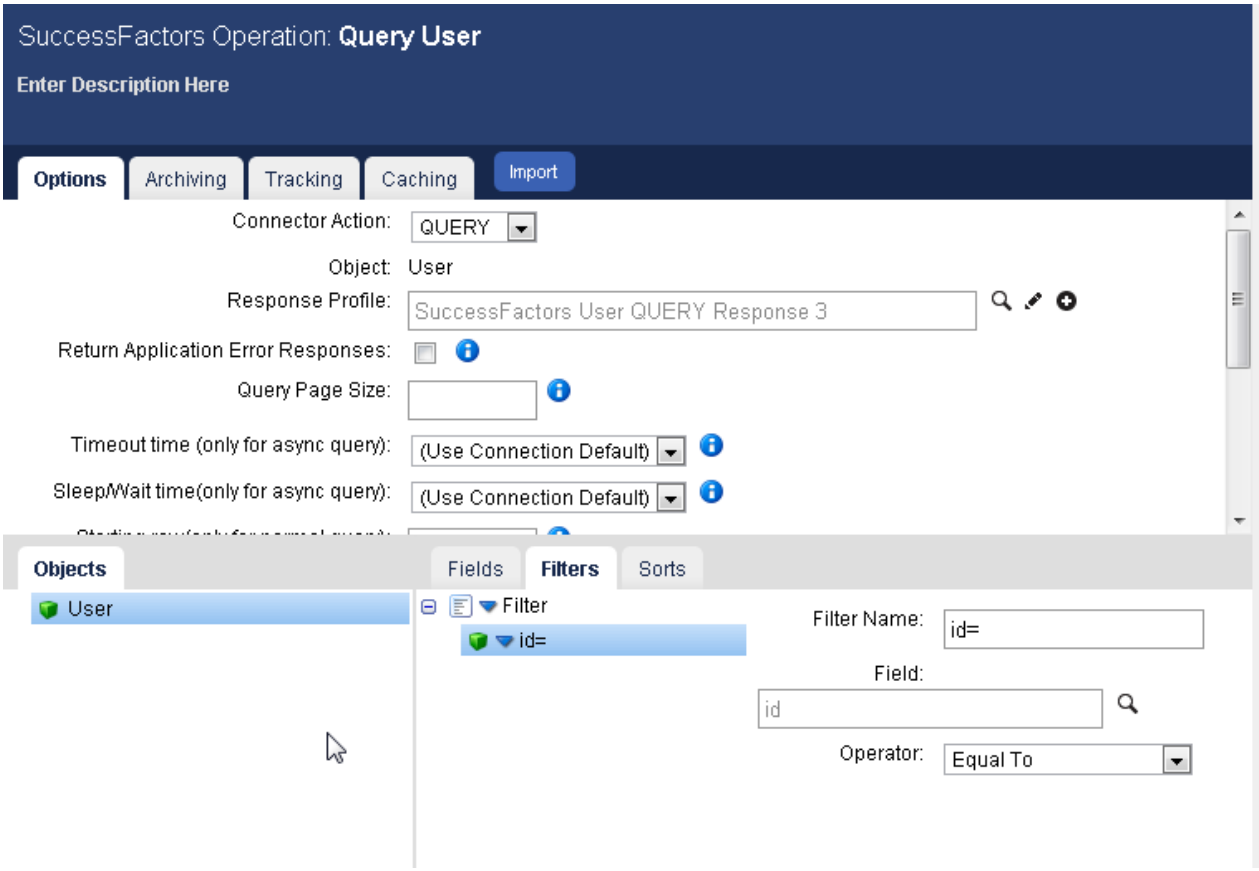


Best Practice: For best performance it is best to select only the fields necessary for the integration process but beware, if you fail to select a field, it will always be blank in the response profile and you will get no indication of that unless the response XML Profile has it flagged as a mandatory field.

> **Note**
>
> If you perform a re-import of an operation, any newly added fields will be automatically selected. It may be necessary to turn off these fields if not used in the integration process.

2.2.2.1.22 Query Filter

This sub tab specifies filter criteria to limit the records returned that conform to a specified search criteria.



> **Note**
>
> The only fields that will appear on the Query Filter sub tab are those fields indicated as *Filterable* or *Constrainable* per the object type specification in the *SFAPI Data Dictionary*.

Both filterable and constrainable fields will support at least one of the following filter operators as specified in the Operator drop down list:

- EQUAL TO - a record's field value matches the specified filter parameter value
- GREATER THAN - a record's field value is greater than the specified filter parameter value
- LESS THAN - a record's field value is less than the specified filter parameter value
- GREATER THAN OR EQUAL TO - a record's field value is greater than or equal to the specified filter parameter value
- LESS THAN OR EQUAL TO - - a record's field value is less than or equal to the specified filter parameter value
- IS NULL - a record's field field is blank
- IS NOT NULL - a record's field field is not blank
- IN - a record's field value matches one of a comma delimited list of values specified in the filter parameter.

> **Note**
>
> The operator drop down list will only include filter operators that field supports per the object type specification in the *SFAPI Data Dictionary*.

> **Note**
>
> Important: Except for IS NULL and IS NOT NULL operators, you must also specify a value for each filter argument on the Connectors Parameters Tab accessible by double clicking on the Connector step.

Best Practice: The IN operator can be used to join the results of several synchronous queries of individual object types. This provides an alternate to using ADHOC reports to join related object types. This technique will result in fast performance, but use with caution with extremely large data sets.

1. First query the set of parent records using filter criteria to limit the number of documents returned.
2. Connect this record to two branches, the first of which builds a comma delimited list of parent ID's from the result set.
3. Feed this comma delimited list to one or more branches connecting to synchronous query operations for each of the child object types of the join, each of which has a filter that uses an IN operator on the foreign keys (parent ID) for each record.
4. Feed the result set of each child query operation to a Document Cache step indexed by the foreign key (parent ID). Note the IN filter parameter will be set to receive the Current Data which contains the comma delimited list of parent IDs.
5. Connect the second branch from the parent query to a map function that uses the parent documents as the main profile, and uses Document Cache Map Lookups to join in each child object type.

> **Note**
>
> An IN operator has a maximum of 1000 values, so use of the Boomi Flow Control step may be required to batch comma separated ID values to a query that utilizes an IN filter operation.

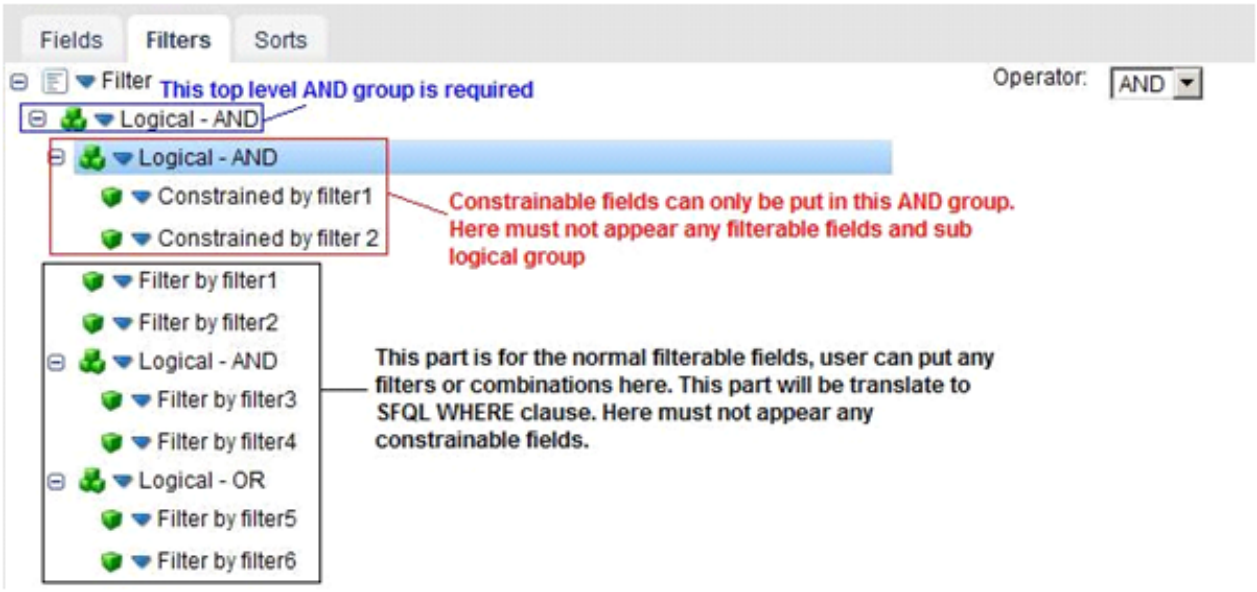
2.2.2.1.23 Filterable vs. Constrainable

Synchronous query operations only support Filterable fields. Filters for such operations can specify complex boolean expressions consisting of nested AND and OR logical operators.

ADHOC_ object types support both Filterable and Constrainable fields. Constrainable fields represent "required fields" for a report filter. These are fields that specify the population of users to include in the query results (the report builder People tab) and required fields for effective dated reports such as effective date ranges to limit the dates of returned records.

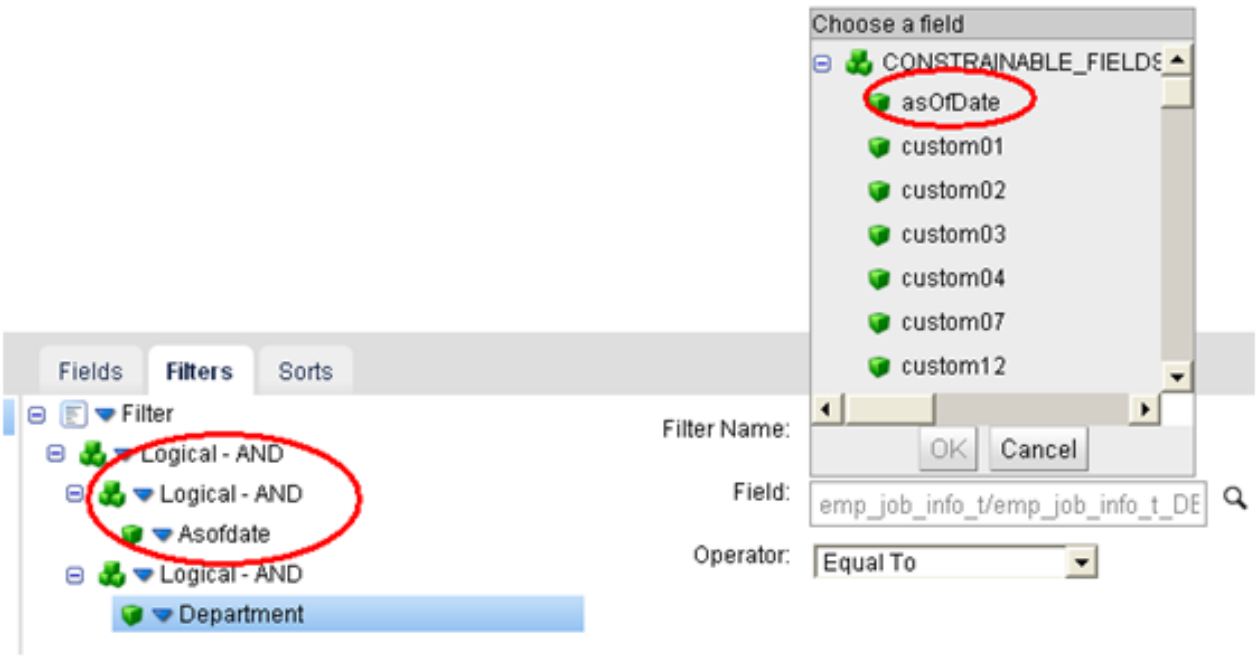
Filters using Constrainable fields must be a single level of filter operators using only the AND logical operator. Neither complex nest expressions nor OR operators are supported.

Use of filters with constrainable fields requires restrictions on the structure of a filter "tree" as displayed in the figure below:



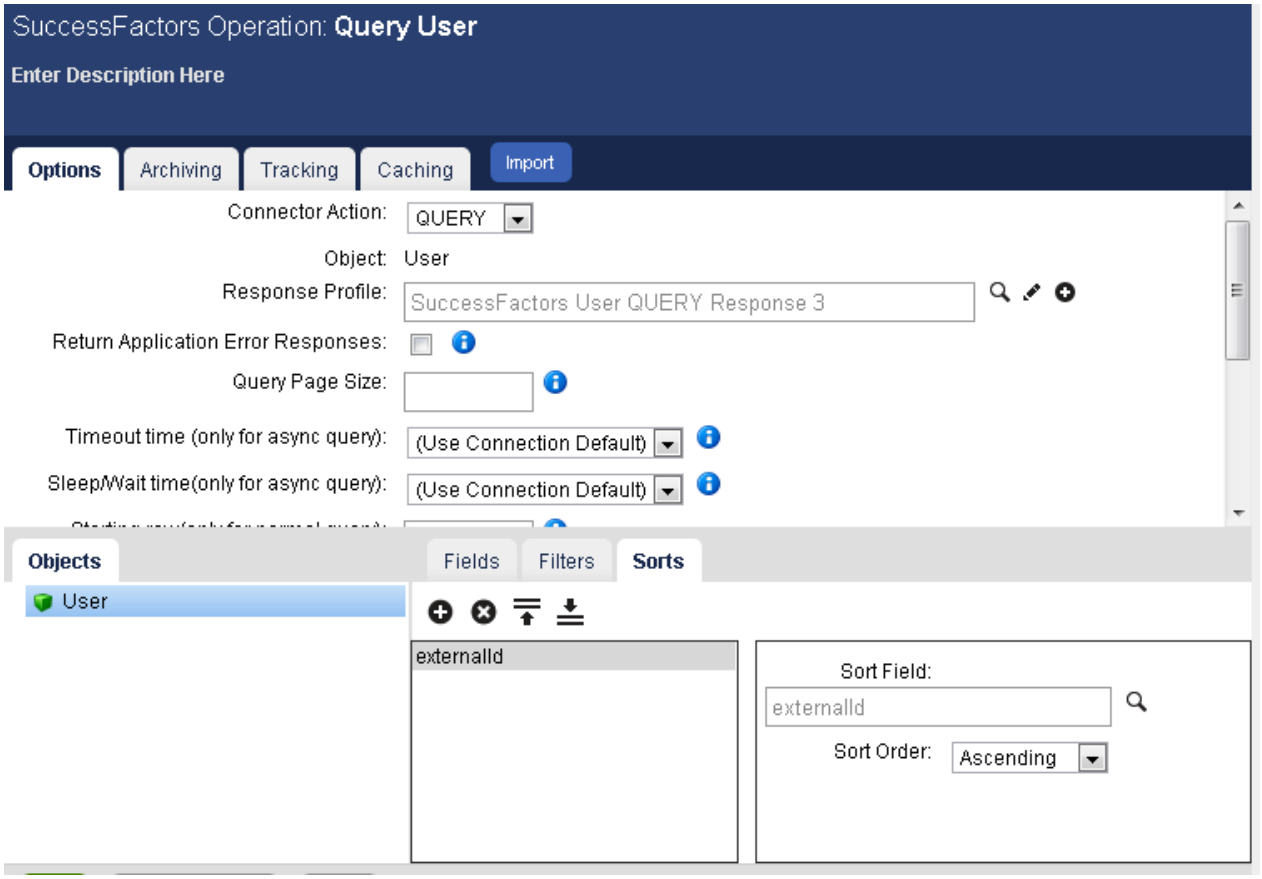
AdhocReport_ object types only provide Constrainable fields for the filter set up by the report designer. No filterable fields are exposed.

An example of of a setting up filter using Constrainable fields is displayed below. This filter specifies the "Snapshot date" to limit the values of effective dated records to values that are in effect on the specified date.



2.2.2.1.24 Query Sort

This sub tab enables sorting the results of a query by the specified fields values. Sortable fields support either Ascending or Descending sort order.

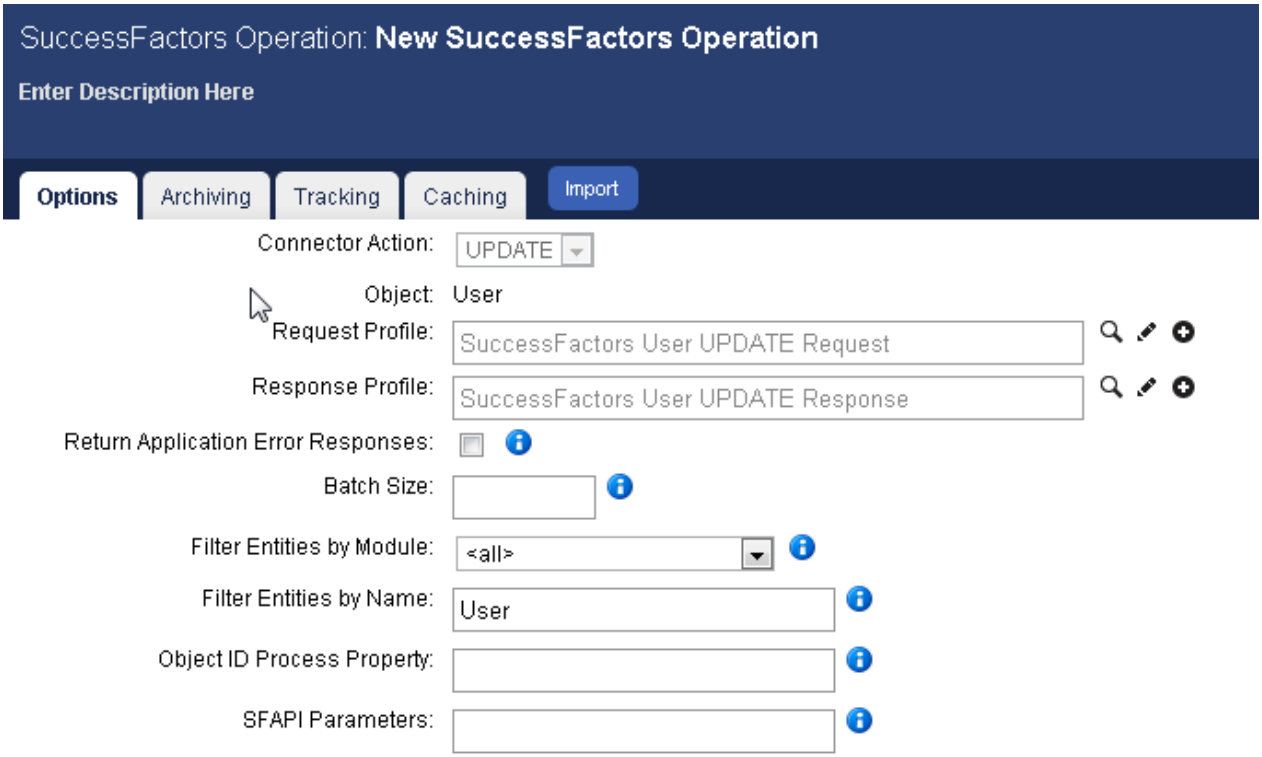


> ### Note
>
> The only fields that will appear on the Query Sort sub tab are those fields indicated as *Sortable* per the object type specification in the *SFAP Data Dictionary*.

Best Practice: Open the Boomi log file to view the raw query statement for the executed query operation. Simply scroll down to where the "SELECT " statement appears.

2.2.2.2 Write Operations

Write operations are those that perform the Create, Update, Upsert or Delete actions. For more information regarding SFAPI specifics of these actions, please refer to SAP SuccessFactors HCM suite SFAPI: Developer Guide.



2.2.2.2.1 Connector Action

Indicates the type of write operation.

> **Note**
>
> If you change the Connector Action, a new import will be required and the existing settings will be deleted.

2.2.2.2.2 Object

Defines the object on which the operation is executing. This is only set during an import operation.

2.2.2.2.3 ODATA Method/Operation

This specifies the type of ODATA write operation that will be performed. User can choose from Delete, Update/Merge, Update/Replace, Create/Post, Upsert/Post.

2.2.2.2.4 Request Profile

The XML profile that defines the XML schema/structure of the response document returned by the operation. For more information regarding profiles, please see *XML Profiles*.

2.2.2.2.5 Response Profile

The XML profile that defines the XML schema/structure of the response document returned by the operation. For more information regarding profiles, please see *XML Profiles*.

2.2.2.2.6 Batch Size

Overrides the write Batch Size specified in the SAP SuccessFactors Connection.

> ⓘ Note
>
> If you override this value, you cannot control it from a Process Extension.

2.2.2.2.7 Return Application Error Responses

Defines whether to return exception responses through the process or fail at the connector level.

Best Practice: Leave *Return Application Error Responses* unchecked and catch the errors using a Try Catch step.

2.2.2.2.8 Filter Entities by Module

For Import Only. This field has no effect for existing operations and will only reflect what value, if any, was used during import.

2.2.2.2.9 Filter Entities by Name

For Import Only. This field has no effect for existing operations and will only reflect what value, if any, was used during import.

2.2.2.2.10 Object ID Process Property

(Optional) Specifies the name of the process property that contains the object type. This overrides the object type ID extracted during Import. This is generally used for Form object types to allow the object type ID to be specified dynamically as an Environment Extension.

Best Practice: you should always use this field for Form object types because the integer component of a Form object type ID (For example, "PMForm$22") can vary from instance to instance.

2.2.2.2.11 SFAPI Parameters

Use this field to specify a comma delimited list of name/value pairs that specify extra parameters available to specific HCM suite object types. For example, a User object type create operation, this field could contain "managerTransfer:true, routeInboxDoc:false" specified as SFAPI parameters to set the `managerTransfer` parameter to "true" and `routeInboxDoc` parameter to "false".

For more information, please refer to the *SFAPI Entity Type Guide*.

3 Setting Up OAuth 2.0 in Boomi Connectors

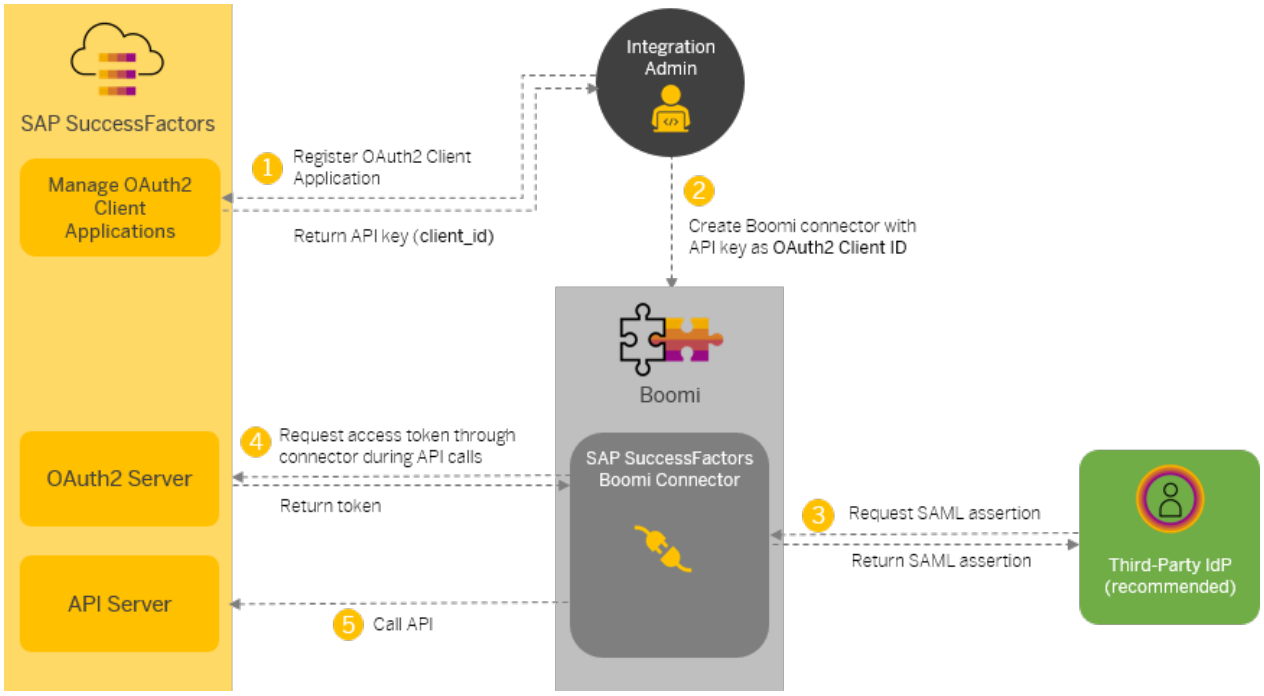
Learn how to set up and use OAuth 2.0 for authentication in Boomi connectors.

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 diagrams explain how OAuth 2.0 works with SAP SuccessFactors Boomi connectors. We recommend that you use a trusted third-party IdP to obtain SAML assertions for authentication.

Option 1: Use Third-Party IdP to Generate SAML Assertions (Recommended)

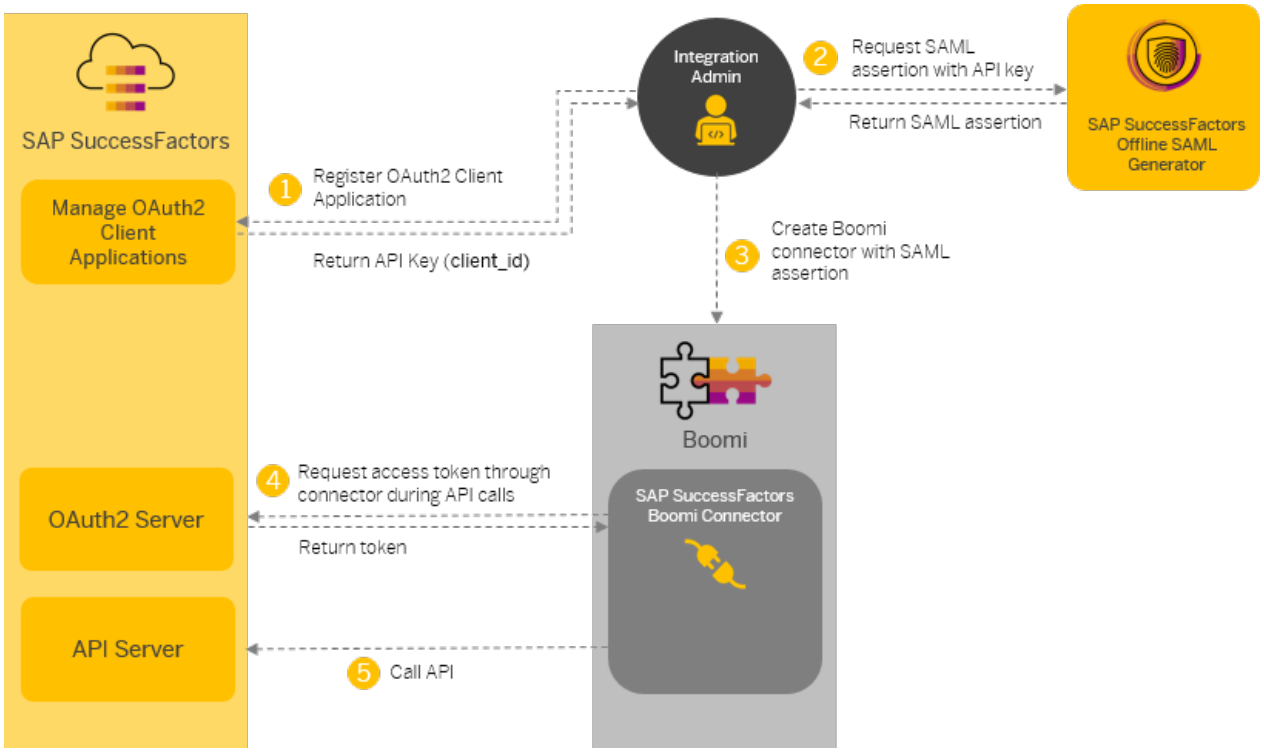


- Registering Your OAuth2 Client Application [page 37]
- Creating a Connection [page 7]
- Some third-party IdPs require additional steps to use their SAML assertions, such as user consent. In that case, you might not be able to fully automate the authentication process. Follow the documentation of that IdP for detailed instructions on how to use the service. [page 35]

1. Register your client application in SAP SuccessFactors to obtain an API key. You'll need the API key in the *OAuth2 Client Id* field when you create a Boomi connector.
2. Create a SAP SuccessFactors connector in Boomi with the API key. Depending on the capabilities of the third-party IdP, you can either set up a process to automatically request SAML assertions from the IdP or manually request a SAML assertion and enter it in the connector.
   - If automation is possible, enter the variable field in *OAuth2 SAML Assertion Field* of the connector used to sstore the SAML assertion.
   - If automation isn't possible, manually request a SAML assertion and enter it in *OAuth2 SAML Assertion* of the connector.

   > ⚠ Caution
   >
   > Some third-party IdPs require additional steps to use their SAML assertions, such as user consent. In that case, you might not be able to fully automate the authentication process. Follow the documentation of that IdP for detailed instructions on how to use the service.
3. In an automated setup, the Boomi connector communicates with the third-party IdP to obtain a SAML assertion.
4. During integration, the connector passes the SAML assertion along with other information to the API server to request an access token.
5. The connector uses the returned access token to call APIs.

Option 2: Use Offline Tool to Generate SAML Assertions



- Registering Your OAuth2 Client Application [page 37]

- Creating a Connection [page 7]
- Generating a SAML Assertion [page 43]

1. Register your client application in SAP SuccessFactors to obtain an API key. You'll need the API key in the *OAuth2 Client Id* field when you create a Boomi connector.
2. Use the offline SAML generator provided by SAP SuccessFactors to generate SAML assertions.
3. Create an SAP SuccessFactors connector in Boomi and enter the SAML assertion in the *OAuth2 SAML Assertion* field.
4. During integration, the connector passes the SAML assertion along with other information to the API server to request an access token.
5. The connector uses the returned access token to call APIs.

For more information, see the Related Information section.

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.

Option	Description
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. **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. **Note** Contact your system administrator or Technical 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 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. You can obtain a certificate from a trusted service provider, or generate a self-signed certificate using a third-party tool. Either way, the certificate must be encrypted using a secure signature algorithm. Although both RSA-1 and RSA-2 algo-

Option	Description
	rithms are supported, we recommend using RSA-2 for better security. If neither option is available, you can also generate an X.509 certificate in SAP SuccessFactors. If you choose to provide For more information, see the *Related Information* section of this topic. **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. **Caution** 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.

3.1.1 Creating a X.509 Certificate Using Your Own Tools

You can use tools such as OpenSSL to create an X.509 certificate.

Prerequisites

There are different tools you can use to create X.509 certificates. In this example, we'll show you how to use OpenSSL to create a certificate. For Windows users, 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, 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. Use the `openssl` command to create an X.509 certificate. The example below shows how to create a certificate using the recommended SHA-2 signature algorithm:

   ```
   $ openssl req -nodes -x509 -sha256 -newkey rsa:2048 -keyout private.pem -out public.pem
   ```

   > **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.
   >
   > Although SAP SuccessFactors support certificates signed using either SHA-1 or SHA-2 algorithms, we recommend that you use SHA-2 for better security.

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++GKyyS8fIhJgeWSb6EelMFhjQ0nZKzbZX5ms3I91n
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 HCM suite if you're unable to create a certificate using your own tools.

Context

> ⚠ Caution
>
> We don't recommend creating the X.509 certificate in the API Center and downloading the private key. It's a less secure approach because downloading the private key increases the risk of exposing it. Only consider this approach you're unable to create an X.509 certificate using your own tools.

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.

Option	Description
	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 37] 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.

Context

You have the following options to generate a SAML assertion:

- (Recommended) Use a corporate IdP, for example, SAP Identity Authentication Services, or a third-party IdP. Refer to the documentation of the corporate IdP for detailed instructions.

> **Note**
>
> Signing certificates using SHA-1 is no longer supported. We recommend that you use SHA-2 for better security. For more information, see the Related Information section.

- Use the example code attached to 3031657 to generate SAML assertions for your application. This sample code provides a SAML generator tool that processes the input information offline and generates a SAML assertion without having to expose your private key to the Internet.

  > **Caution**
  >
  > Any software coding and/or code snippets are examples. If you use any examples to help generate an SAML Assertion that will be used in a production environment, you are solely responsible for ensuring the security of such SAML Assertions. SAP does not warrant the correctness and completeness of the example code and such code is delivered "AS-IS". 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.

> **Note**
>
> Do not use the `/oauth/idp` API to generate SAML assertions. This approach is unsecure and has been deprecated. For more information, see the Related Information section.

Required Elements for IdP-based SAML Assertions

If you choose to use an identity provider (IdP) to generate a SAML assertion, make sure that you follow the SAML 2.0 standard and include the following elements 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 IdP-based SAML Assertions

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

Element	Description	Example

`<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 37].	`<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>`

Element	Description	Example

`<saml2:AttributeStatement>` and `<saml2:Attribute>`	If you want to authenticate SAP SuccessFactors Learning users that don't also exist in platform, you must include an additional custom attribute called "external_user" with value "true".	`<saml2:AttributeStatement>`
		`  <saml2:Attribute Name="external_user">`
		`  <saml2:AttributeValue xsi:type="xs:string">true </saml2:AttributeValue>`
		`  </saml2:Attribute>`
		`</saml2:AttributeStatement>`

Related Information

Deprecation of OAuth IdP API /oauth/idp

Deprecation of SHA-1 Signing Certificate for Assertion Consumer Services (ACS) Entries

4 Tools and References

A list of tools and reference documentation available in SAP SuccessFactors.

The following tools are available in the Admin Center in your SAP SuccessFactors instance:

- Legacy SFAPI Data Dictionary
- Legacy SFAPI IP Whitelisting
- SFAPI Audit Log
- SFAPI Metering Details

You can refer to the SAP SuccessFactors HCM suite SFAPI: Developer Guide for more details of how to use SFAPIs.

5 Change History

Learn about changes to the documentation for SAP SuccessFactors HCM suite Boomi Connector Reference Guide in recent releases.

2H 2023

Type of Change	Description	More Info
Changed	We moved the Change History to the end of the guide.	Introduction [page 3]

2H 2021

Type of Change	Description	More Info
Changed	We updated the OAuth 2.0 topic to include separate process flows for generating SAML assertions using a third-party IdP and the offline tool.	Setting Up OAuth 2.0 in Boomi Connectors [page 35]

1H 2021

Type of Change	Description	More Info
New	Added support for OAuth 2.0	Creating a Connection [page 7] Authentication Using OAuth 2.0
Changed	We updated the topics where instance-level and API-specific IP restrictions and exceptions apply.	Requesting an Access Token
Changed	We updated the supported signing algorithms and our recommendation for generating SAML assertions using a third-party IdP. We also added the required elements in the SAML assertions.	Generating a SAML Assertion [page 43]

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.

www.sap.com/contactsap

THE BEST RUN SAP