



SAP SuccessFactors 

PUBLIC

Document Version: 2H 2023 – 2024-04-01

SAP SuccessFactors HXM Suite SFAPI: Developer Guide

Content

1	Introduction.	4
1.1	List of SAP SuccessFactors API Servers.	5
2	Deprecation of Partner API, SFAPI Adhoc, and SFAPI for Simple Entities.	10
2.1	API Deprecation FAQs.	11
3	Using SFAPI.	15
3.1	Permissions.	15
3.2	Authenticating SFAPI Users.	15
	Authentication Using OAuth 2.0.	19
3.3	Restricting API Access by IP Addresses or IP Address Ranges.	32
3.4	Integration Tools.	33
3.5	API Thresholds and Limitations.	35
4	SFAPI Operations.	37
4.1	API Summary.	37
4.2	Session Management Operations.	39
4.3	MetaData Discovery Methods.	44
4.4	Data Manipulation Operations.	54
4.5	Data Query Operations.	67
4.6	Async Job Operations.	74
5	User Entity.	84
5.1	Overview.	84
5.2	Permissions.	84
5.3	Field Configuration.	85
5.4	Differences between API and File Import/Export of Employee Data.	85
5.5	Business Rules for Data Transfer.	86
5.6	Common Fields in HRIS Integration.	90
5.7	Optional Business Processing Parameters.	96
5.8	Example Upsert With SFPParameter.	100
5.9	Time Zone Values.	101
5.10	Deprecated Time Zone Values.	106
5.11	Supported Countries and Regions.	106
5.12	Passwords.	109
6	SFAPI Error Messages.	110
6.1	Calibration Module Error Messages.	110

- 6.2 Common Module Error Messages. 110
- 6.3 CDP Module Error Messages. 114
- 6.4 RCM Module Error Messages. 116
- 6.5 User Module Error Messages. 117

- 7 New DateTime Fields in Recruiting Entities. 123**

- 8 SFAPI Best Practices. 125**

- 9 Change History. 127**

1 Introduction

This document is the handbook for SFAPI developers. This guide provides an overview of the SFAPI, technical information on how to use the SFAPI, details of the Web Service methods and the framework objects.

Background

The SFAPI is SAP SuccessFactors Data API. It is a SOAP Web Service designed for importing and exporting data to and from your SAP SuccessFactors instance. It provides generic create, read, update and delete operations to access data, as well as meta-data operations to allow runtime discovery of the data.

Data are exposed as entities called SFObjects, which are conceptually analogous to database tables. Using the meta-data operations, you can list the SFObjects available to the API, and describe the fields in these entities. Using the CRUD operations you can query or edit the data.

The combination of CRUD style data access operations along with meta-data driven operations for data discovery provides a generic API that offers the following benefits:

- Provide a consistent mechanism for accessing data in SAP SuccessFactors.
- Provide a mechanism to describe the data schema configuration, including the entities that are available and fields that appear in these entities.
- Avoid WSDL changes with each release.

API Endpoints

Your API endpoint will depend on where your SAP SuccessFactors instance is located. It can be in one of several data centers. Please contact your SAP SuccessFactors representative if you are unsure of which data center to use. You'll find the endpoints in the Related Information section.

Related Information

[List of SAP SuccessFactors API Servers \[page 5\]](#)

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/
REST	/rest/	https://api17.sapsf.com/rest/
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 and mTLS certificate servers for SAP SuccessFactors data centers. Use search and filter to find the corresponding servers for your company.

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](#).

Data Center	Environment	Platform	Location	API Server	mTLS Certificate Server
DC2 (DC57)	Production	Google Cloud Platform	Eemshaven, The Netherlands	https://api2.successfactors.eu/	https://api2.cert.successfactors.eu
DC2 (DC57)	Sales Demo	Google Cloud Platform	Eemshaven, The Netherlands	https://apisalesdemo2.successfactors.eu/	https://apisalesdemo2.cert.successfactors.eu

Data Center	Environment	Platform	Location	API Server	mTLS Certificate Server
DC2 (DC57)	Preview	Google Cloud Platform	Eemshaven, The Netherlands	https://api2preview.sapsf.eu/	https://api2preview.cert.sapsf.eu
DC4 (DC68)	Production	Azure	Chandler, Arizona, US	https://api4.successfactors.com/	https://api4.cert.successfactors.com
DC4 (DC68)	Preview	Azure	Chandler, Arizona, US	https://api4preview.sapsf.com/	https://api4preview.cert.sapsf.com
DC8 (DC70)	Production	Azure	Ashburn, Virginia, US	https://api8.successfactors.com/	https://api8.cert.successfactors.com
DC8 (DC70)	Sales Demo	Azure	Ashburn, Virginia, US	https://apisalesdemo8.successfactors.com/	https://api8sales.cert.successfactors.com
DC8 (DC70)	Preview	Azure	Ashburn, Virginia, US	https://api8preview.sapsf.com/	https://api8preview.cert.sapsf.com
DC10 (DC66)	Production	Azure	Sydney, Australia	https://api10.successfactors.com/	https://api10.cert.successfactors.com
DC10 (DC66)	Preview	Azure	Sydney, Australia	https://api10preview.sapsf.com/	https://api10preview.cert.sapsf.com
DC12 (DC33)	Production		Rot, Germany	https://api012.successfactors.eu/	https://api012.cert.successfactors.eu

Data Center	Environment	Platform	Location	API Server	mTLS Certificate Server
DC12 (DC33)	Preview		Rot, Germany	https://api12preview.sapsf.eu/	https://api12preview.cert.sapsf.eu
DC15 (DC30)	Production		Shanghai, China	https://api15.sapsf.cn/	https://api15.cert.sapsf.cn
DC15 (DC30)	Preview		Shanghai, China	https://api15preview.sapsf.cn/	https://api15preview.cert.sapsf.cn
DC17 (DC60)	Preview	Azure	Toronto, Canada	https://api17preview.sapsf.com/	https://api17preview.cert.sapsf.com
DC17 (DC60)	Production	Azure	Toronto, Canada	https://api17.sapsf.com/	https://api17.cert.sapsf.com
DC19 (DC62)	Preview	Azure	Sao Paulo, Brazil	https://api19preview.sapsf.com/	https://api19preview.cert.sapsf.com
DC19 (DC62)	Production	Azure	Sao Paulo, Brazil	https://api19.sapsf.com/	https://api19.cert.sapsf.com
DC22	Preview		Dubai, UAE	https://api22preview.sapsf.com/	https://api22preview.cert.sapsf.com
DC22	Production		Dubai, UAE	https://api22.sapsf.com/	https://api22.cert.sapsf.com
DC23	Preview		Riyadh, Saudi Arabia	https://api23preview.sapsf.com/	https://api23preview.cert.sapsf.com

Data Center	Environment	Platform	Location	API Server	mTLS Certificate Server
DC23	Production		Riyadh, Saudi Arabia	https://api23.sapsf.com/	https://api23.cert.sapsf.com
DC40	Sales Demo	Azure		https://api40sales.sapsf.com	https://api40sales.cert.sapsf.com
DC41	Preview	Azure	US East	https://api41preview.sapsf.com	https://api41preview.cert.sapsf.com
DC41	Production	Azure	US East	https://api41.sapsf.com	https://api41.cert.sapsf.com
DC44 (DC52)	Preview	Google Cloud Platform	Singapore	https://api44preview.sapsf.com/	https://api44preview.cert.sapsf.com
DC44 (DC52)	Production	Google Cloud Platform	Singapore	https://api44.sapsf.com/	https://api44.cert.sapsf.com
DC47	Preview		Canada Central (Microsoft Azure)	https://api47preview.sapsf.com/	https://api47preview.cert.sapsf.com
DC47	Production		Canada Central (Microsoft Azure)	https://api47.sapsf.com/	https://api47.cert.sapsf.com
DC50	Preview	Google Cloud Platform	Aisa Northeast, Tokyo (Google Cloud Platform)	https://api50preview.sapsf.com	https://api50preview.cert.sapsf.com
DC50	Production	Google Cloud Platform	Aisa Northeast, Tokyo (Google Cloud Platform)	https://api50.sapsf.com	https://api50.cert.sapsf.com

Data Center	Environment	Platform	Location	API Server	mTLS Certificate Server
DC55	Preview	Google Cloud Platform	Europe West 3	https://api55preview.sapsf.eu/	https://api55preview.cert.sapsf.eu
DC55	Production	Google Cloud Platform	Europe West 3	https://api55.sapsf.eu/	https://api55.cert.sapsf.eu
DC4 (DC68)	Sales Demo	Azure	US East 2	https://api68sales.successfactors.com	https://api68sales.cert.successfactors.com/
DC74	Preview	Azure	Zurich, Switzerland	https://api74preview.sapsf.eu	https://api74preview.cert.sapsf.eu
DC74	Production	Azure	Zurich, Switzerland	https://api74.sapsf.eu	https://api74.cert.sapsf.eu

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

SAP provides public APIs to developers to access SAP services so that they can extend our solutions, integrate them with other solutions, and build new ones. Please be advised that the following APIs have been deprecated in favor of OData API:

- The Partner/SOAP API
- The SOAP SFAPI for Adhoc Reports
- All other SOAP SFAPIs for simple entity/object types, except for `CompoundEmployee`. Currently, there is no deprecation announcement for `CompoundEmployee`.

Note

The SOAP SFAPIs for Employee Central, except the `CompoundEmployee` API, are being deprecated and will be deleted. For more information, refer to *Deprecation of Employee Central SFAPIs Except CompoundEmployee API* in the SAP SuccessFactors What's New Viewer.

What is API deprecation?

Deprecation is an industry standard terminology for the initial step of the API end-of-life cycle. After deprecation. No enhancements or new features will be delivered for the API. Deprecated APIs may be turned off in newly created tenants by default, but can still be turned on in Provisioning. Deprecation is not retirement although retirement is a possible later stage. It does not mean that an API will be removed from productive environments.

What does API deprecation mean for me?

The stated SFAPIs should not be used for new development as they will be turned off in new tenants by default. Please note that SFAPI can still be enabled in a tenant via provisioning and Instance Sync will copy API settings to new tenants. Starting immediately, we encourage the use of new APIs which leverage Open Data Protocol (OData) technology. These new APIs are available in Provisioning.

What can I do now?

Please use the new OData technology for any new API development and consider moving your legacy applications and integrations to OData API. You can refer to <https://help.sap.com> for more information about OData API.

If you are going to refactor old integrations, please consider using the [Integration Center](#) to realize lower cost of ownership.

We also recommend that you review the API Deprecation FAQs section of this document.

→ Remember

As a customer, you don't have access to Provisioning. To complete tasks in Provisioning, contact your implementation partner or Account Executive. For any non-implementation tasks, contact Product Support.

2.1 API Deprecation FAQs

What is an API?

An API is a publicly available programming interface that is exposed as a standard http transaction. In a nutshell, the only difference between a web page and a web API is that the API is called by a machine while the web page is accessed by a human.

APIs are used when customers build custom SAP Cloud Extensions, when customers are building integrations with CPI, Dell Boomi, and other general cases.

What is the Partner API?

The partner API is an old SOAP API available in our DC4 and DC8 data centers. It has not been documented for more than 5 years.

What is the SFAPI Adhoc Reports API?

These are very inefficient APIs that rely on creating an adhoc report in our system, scheduling the execution of the report on our scheduler servers, and waiting for the job to complete. Once complete they provide the ability to download a CSV file containing the data.

What is SFAPI for "Simple Entities"?

These are APIs that allow access to single entities such as `Person`, `Employment`, etc. Unlike `CompoundEmployee`, they cannot provide complex related employee data.

What is SFAPI for CompoundEmployee?

This is an advanced API built on the SFAPI protocol that allows access to a complete set of entities related to the `Person` entity.

What is API Deprecation?

API deprecation is an announcement that the API should not be used for new development and that SAP reserves the right to decommission it in the future.

What is the difference between deprecation and retirement?

- Deprecation is an announcement. When we deprecate an API, we may have it turned off by default or even disallow it in new instances.
- Decommission means the API will be disabled in a tenant; it will no longer be available in productive environments. At this point, any usage of the API will result in a failure.

Why deprecate an API?

- Discourages partners and customers from investing new development on old technology when newer capabilities are available in OData.
- Our new APIs offer more benefits to customers.
- Allows us to focus on improving OData, not maintaining old APIs.

Will the new APIs, which leverage Open Data Protocol (OData) technology, offer added benefits?

Yes. Some of the benefits include:

- OData is the only API for newer functionality including Position Management, CRM, Simplify, Time, Benefits, Job Profile.
- Basic OData queries are faster than older SFAPI single-entity APIs. For example, we measured a basic, `PerPerson` query up to four times faster using OData.
- OData's new snap shot query feature solves pagination problems that exist in SFAPI.
- OData offers more flexibility in joining related data.
- OData is the only API available in [Integration Center](#).
- Unlike SFAPI-Adhoc, OData does not rely on creating and running reports on a scheduler server. There is no risk of a scheduler being backed up, etc.

Why is OData better than adhoc SFAPI?

When you run an adhoc report API, it doesn't simply run directly against the data as OData does. Instead, it does the following:

1. Creates a new report based on the SFAPI query provided.
2. Schedules the report to run on the scheduler server.
3. Client has to iteratively poll the server to wait to check if the report is completed.
4. If and when the report execution completes, the client can then download the results in CSV. If the scheduler queue becomes backed up, the report might not finish in time and the client will time out and not wait longer.

How can I move from SFAPI to OData?

1. Leverage our documentation at help.sap.com:
 - [SAP SuccessFactors HCM suite Boomi Connector Guide](#)
 - [SAP SuccessFactors HCM suite OData API: Developer Guide](#)
 - [Implementing the Employee Central Compound Employee API](#)
2. Learn about the OData capabilities of SAP CPI and/or Dell Boomi.
3. Consider reimplementing integrations using [Integration Center](#). You will find integrations are easier to develop, maintain and will have lower cost of ownership. You may also find this easier than porting simpler legacy integrations to OData.

Will the API be removed from productive environments once deprecated?

No. This will only occur when the API is retired and that will occur no sooner than 1 year from the deprecation announcement date.

Is the SFAPI listed in the API Center (Admin Center) as deprecated?

Yes, it is listed.

If an API is deprecated, is it unsupported?

No. We'll continue to support SFAPI and provide bug fixes during the deprecation period. However, we'll no longer develop new functionalities for deprecated APIs, and all support will be stopped when an API reaches the end of maintenance date.

How are APIs deprecated?

API deprecation can occur on a single function or data entity as well as an entire class/protocol of APIs.

How will I know if my tenant is using deprecated APIs?

You can run the SFAPI feature status check (Check ID: SFAPIsTurnedOff) in the Check Tool under the ► [System Health](#) ► [Application](#) ► [API](#) category to see whether SFAPI is turned off in your instance. For more information about the Check Tool, see [Using the Check Tool](#).

What is our policy around APIs?

APIs that have been used for more than 24 months may be deprecated to improve the quality and functionality of our APIs. Our API policy states that:

- We will not deprecate an API sooner than 2 years after initial release.
- We will not stop supporting or retire an API sooner than 1 year after deprecation.
- We may announce the retirement sometime after deprecation and at least 1 year prior to the retirement date.

📌 Note

We understand your reliance on APIs, so we will evaluate the usage of the APIs before decommissioning them. The primary goal of the deprecation is to stop new development on old APIs and encourage moving to OData.

3 Using SFAPI

3.1 Permissions

List of permissions required for users to access SFAPI.

Users must have the API login permission and data specific permissions to access SFAPI. As a company administrator, grant the following permissions to your SFAPI users:

- [▶ General User Permissions ▶ SFAPI User Login ▶](#): This permission allows to log in to SFAPI. This permission should only be granted to administrators for integration purpose.
- In addition to [SFAPI User Login](#) permission, there are permissions that control access to data in SFAPI. These permissions vary depending on the data being accessed.
Access to different objects in the API requires the related permissions. Authorizations in SFAPI are administrative in nature. This means that the permissions required to access data through the SFAPI are appropriate for administrators, granting access to broad sets of data. These permissions aren't appropriate for business users.
The administrative permissions are useful for data integrations between systems. However, SFAPI doesn't provide use cases appropriate for nonadministrative access (that is, business users). Refer to the SFAPI Data Dictionary to understand the permissions required for various API operations on each entity.

3.2 Authenticating SFAPI Users

Learn how to authenticate SFAPI users.

Prerequisites

SFAPI follows the instance-level IP restriction setting. Before you access an SFAPI, make sure that the client IP address is on the allow list to access the corresponding API server.

SFAPI also follows the API login exceptions set on the [Password & Login Policy Settings](#) page. Make sure the client IP address is also allowed in the setting.

For more information about IP restriction management and login exceptions, see the Related Information section.

Context

Authentication

Authentication is established through login. A successful login returns a session ID as an HTTP Cookie. This cookie must be passed back to the API server in all subsequent HTTP requests in order to authenticate the user. An API session times out after 10 minutes of inactivity. You can manually invalidate a session by logging out.

SFAPI supports both HTTP Basic Authentication and OAuth 2.0. In this topic, you'll learn how to authenticate users using each authentication method.

Note

HTTP Basic Authentication has been deprecated. We recommend that you use OAuth 2.0 to authenticate API users for better security. OAuth 2.0 is recommended if you're building custom integrations using SF APIs. For more information about OAuth 2.0, see the Related Information section.

Handling HTTP Cookies

SFAPI is state-aware, and there are several cookies that must be handled. One is for the login session as noted above. Another cookie is for load balancing in the data center. In the future, there may be other cookies that must be handled. API clients must be able to handle cookies.

Procedure

- Follow the steps below to authenticate users using OAuth 2.0:
 - a. Obtain an OAuth token. For more information, see [Authentication Using OAuth 2.0 \[page 19\]](#).
 - b. Send a POST request with the following information:

HTTP Method	POST
URI	<code>https://<api-server>/sfapi/v1/soap/</code>
Header	<code>Authorization: Bearer <OAuth Token></code>

Request Body

```
<soapenv:Envelope
  xmlns:soapenv="http://
schemas.xmlsoap.org/soap/
envelope/"

xmlns:urn="urn:sfobject.sfapi.success
factors.com">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:login>
      <urn:credential>
        <urn:companyId></
urn:companyId>
        <urn:username></
urn:username>
        <urn:password></
urn:password>
      </urn:credential>
    </urn:login>
  </soapenv:Body>
</soapenv:Envelope>
```

Note

The request body is required and the values of company ID, username, and password fields must be left empty.

Sample response:

```
<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <loginResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
      <result>
        <sessionId>D91E1738628E***6FB46E2.mo-75feed8b0</sessionId>
        <msUntilPwdExpiration xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance" xsi:nil="true" />
      </result>
    </loginResponse>
  </S:Body>
</S:Envelope>
```

c. Pass the session ID in the header of subsequent API calls:

Cookie: JSESSIONID=**D91E1738628E***6FB46E2.mo-75feed8b0**

- Follow the steps below to authenticate users using HTTP Basic Authentication:
 - Send a POST request to the API server with login credentials:

HTTP Method	POST
URI	https://<api-server>/sfapi/v1/soap/
Header	Content-Type: text/xml; charset=UTF-8

Request Body

```
<soapenv:Envelope
xmlns:soapenv="http://
schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.success
factors.com">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:login>
      <urn:credential>

    <urn:companyId>COMPANY</
urn:companyId>

    <urn:username>userabc</urn:username>

    <urn:password>Abc123</urn:password>
      </urn:credential>
    </urn:login>
  </soapenv:Body>
</soapenv:Envelope>
```

Sample response:

```
<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <loginResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
      <result>
        <sessionId>13ADE15B996A90FFAC69303EFD5ED75C.mo-f064bf05b</
sessionId>
        <msUntilPwdExpiration>7859760182</msUntilPwdExpiration>
      </result>
    </loginResponse>
  </S:Body>
</S:Envelope><?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <loginResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
      <result>
        <sessionId>13ADE15B996A***69303EFD5ED75C.mo-f064bf05b</
sessionId>
        <msUntilPwdExpiration>7859760182</msUntilPwdExpiration>
      </result>
    </loginResponse>
  </S:Body>
</S:Envelope>
```

- b. Pass the session ID in the header of subsequent API calls.

Cookie: JSESSIONID=13ADE15B996A***69303EFD5ED75C.mo-f064bf05b

Related Information

[Authentication Using OAuth 2.0 \[page 19\]](#)

[Deprecation of HTTP Basic Authentication](#)

[IP Restrictions](#)

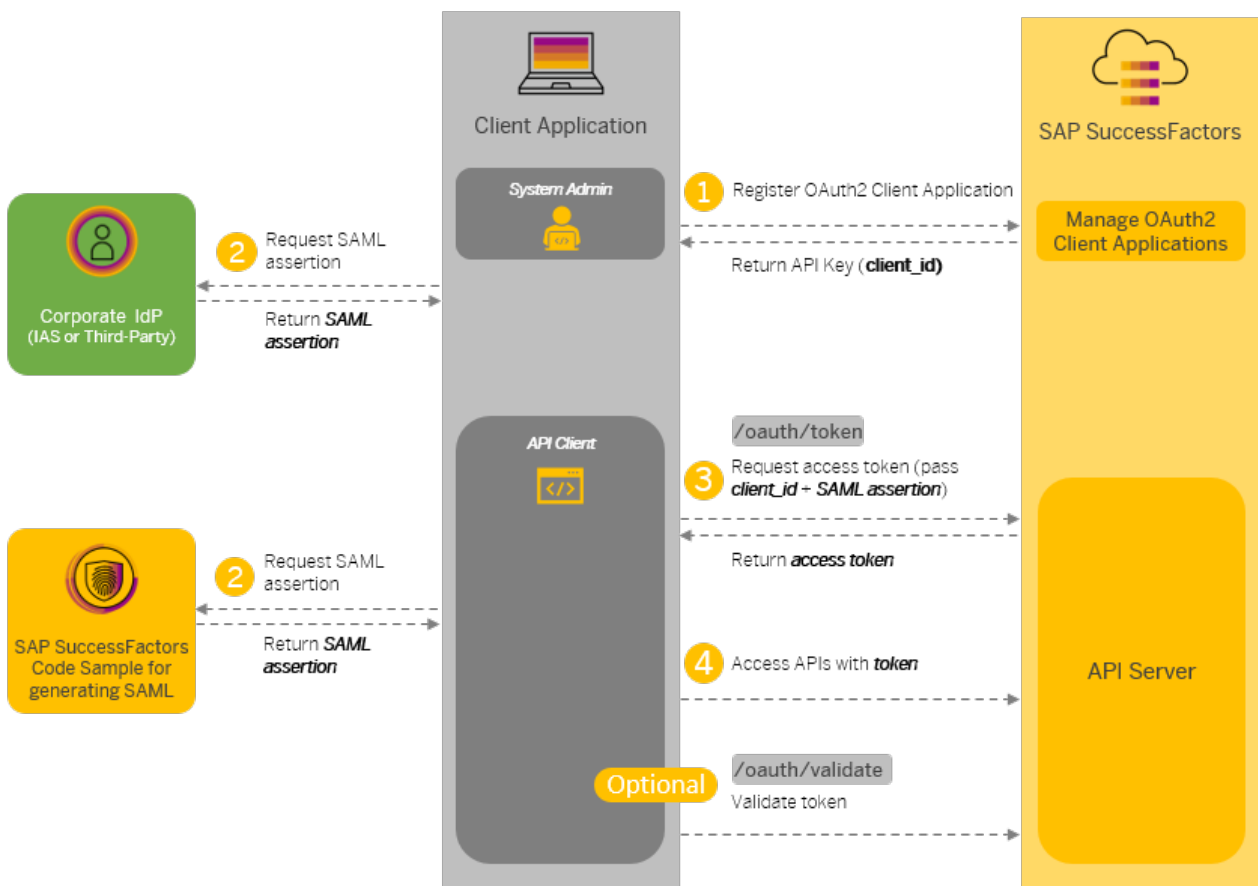
3.2.1 Authentication Using OAuth 2.0

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 \[page 20\]](#)
- [Generating a SAML Assertion \[page 26\]](#)

- [Requesting an Access Token \[page 29\]](#)
 - [Viewing the Validity of an Access Token \[page 31\]](#)
 - [Follow the documentation of the IdP for requesting SAML assertions. \[page 19\]](#)
1. Register your client application in SAP SuccessFactors to obtain an API key.
 2. Obtain a SAML assertion from your trusted IdP (recommended, for example, SAP Cloud Identity Services - Identity Authentication) or use the sample code to generate one. For more information on how to use SAP Cloud Identity Services as an IdP, see the Related Information section.
 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.

Related Information

[SAP Cloud Identity Services - Identity Provisioning](#)

3.2.1.1 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.

Option	Description
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	<p>(Optional) You can enable this option to restrict the access of the application to specific users including business users and technical users.</p> <div data-bbox="841 558 1432 911" style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>Note</p> <p>A business user in this context is a user who has permissions to call SAP SuccessFactors APIs for integration purposes.</p> <p>A technical user is a system-generated user created for integrating SAP SuccessFactors with other SAP products and solutions.</p> <p>Refer to About Technical User for more information.</p> </div>
User IDs	<p>(Required if you enabled the Bind to User option) Enter the user IDs separated by comma.</p> <p>The binding of business users and technical users works as follows:</p> <ul style="list-style-type: none"> • 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. <div data-bbox="841 1514 1432 1675" style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> <p>Note</p> <p>Contact your system administrator or Product Support if you don't know the technical user ID of your instance.</p> </div>
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.

Option

Description

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 algorithms 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.

3.2.1.1.1 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+gAwIBAgIUUKR82LgtkNBccdyPYD26K87zZ+vYwDQYJKoZIhvcNAQEE
BQAwDTELMAkGA1UECwwCRVAvHhcNMTkwOTI2MDIwNDUyWhcNMTkwMDI2MDIwNDUy
WjANMQswCQYDVQQLDAJFUDCBnzANBgkqhkiG9w0BAQEFAAOBjQAwYkCgYEAwKva
NZCOGcuY90/BudS+qQic+A3luM8mLtmI60R1iEjgEWGBCxSiDb2h8mQJiXwku19W
ebaaZP7hkqkdNoJgV/6NE7++GKyyS8fIhJgeWSb6Ee1MFhjQ0nZKzbZX5ms3I91n
twzkcHtKCQi/gi/Rouh1k/P/QVcrzSgHUHQJNY0CAwEAAANTMFEwHQYDVR0OBBYE
FHHbgqnnhm3GAJ4gy2IuEDxpLye7MB8GA1UdIwQYMBaAFHHbgqnnhm3GAJ4gy2Iu
EDxpLye7MA8GA1UdEwEB/wQFMAMBAf8wDQYJKoZIhvcNAQEEBQADgYEAG5CoqcEy
15vUpj5VfJeR/DS70tPIinp/TCC9kRO/+TsnPbqVcfPr8vIyc4L3MPKjXFBsefE
vtfHGGucVtv5N1+4U/b9NxFbuH2MP7W3swZ4WM72Na+W6iohwesOr0p3IcOfxc3
RNCnagFmtbDFxAlPXQ0d+m+N5gxLRoCX1hE=
-----END CERTIFICATE-----
```

Example of a private key:

```
-----BEGIN PRIVATE KEY-----
MIICQDIBADANBgkqhkiG9w0BAQEFAASCAL8wggJbAgEAAoGBAMCr2jWQjhnLmPdP
wnUvqkInPgN9bjPjI7ZiOtedYhI4BFhgQsUog29ofJkCY18JLpfVnm2msz+4ZKp
HTaCYff+jRO/vhiSSkvHyISYH1km+hHpTBY0NJ2Ss22V+ZrNyPdZ7cM5HB7SgkI
v4Iv0aLoZZPz/0FXK80oB1B6iTctAgMBAAEcGyAid5vVsUJ6gt2egHobkF97Rbsu
9PBcW1JtVyUTUW/1LYRIF7VKEirbYm0y04spOTgozxlDMLmIqqAX6ID9W114kN/g
lzlc2/jMg+YGp+FNCjULyggfIwtGfpX8G0qYWza5oarZVbbGAlcvpHjyNMGV7ure
7syrjIXUighkaKrxgQJBAObVbGTVr/5xxScB1mPYoBe02JMyTzuVW0ts7NyfxXJu
w9vUoMDLV+2wuDE4w8/gUkKf26eojn3kwD708V61G4kCQQDVRVC7HcXYfU4wkr5S
JPMQzAln0Ruf6LgFpgIDPKpq7VUt1lA9aQUbdddxcudFj057ksr2yU9sOLQgh3A
+2GFAkAwkRDavsVI48h5asWR11C3YJe3tDhow848DncNjpUX/dop+JyKnJaJBzjK
nxkNjomcN9Kajnd3v9BH11ytewi5AkA8IAWscUc/kJrUziXhpWYD3vXykYG5Ndm6
```



```

NSkx0dmLprZifNSlB7nAyduqqXTe4eVyNxxN3d9PyZs5ArPuno2lAkAQ8WiHbqGA
JlO6R9+D6HiWywpCaQ0oh6H/+84mb1ew2SUw1mFxROxgfsRVNUe+ahs3nSIhoba0
cqS0ZSBtNDxV
-----END PRIVATE KEY-----

```

3.2.1.1.2 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

Option	Description
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.
	<div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 5px;"> <p>Note</p> <p>Validity check works only when the <i>Enable validity check</i> option is selected.</p> </div>
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 <i>Validity</i> 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 \[page 20\]](#) 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.

3.2.1.2 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

Both SHA-2 and SHA-1 signing algorithms are supported. However, we recommend that you use SHA-2 for better security.

- 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.

→ Remember

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.

⚠ Caution

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
<code><saml2:Issuer></code>	Issuer information of the SAML assertion	<code><saml2:Issuer>www.myidp.com</saml2:Issuer></code>

Element	Description	Example
<p><saml2:Subject>, <saml2:NameID>, and Recipient</p>	<p>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.</p>	<pre><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:SubjectConfirmationData> </saml2:SubjectConfirmation> </saml2:Subject></pre>
<p><saml2:AttributeStatement> and <saml2:Attribute></p>	<p>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 20].</p>	<pre><saml2:AttributeStatement> <saml2:Attribute Name="api_key"> <saml2:AttributeValue xsi:type="xs:string">NDU0MDE0MDkwYj***5YTE5MwIxMTNkNjc1Zg</saml2:AttributeValue> </saml2:AttributeValue> </saml2:Attribute> </saml2:AttributeStatement></pre>
<p><saml2:Conditions>, NotBefore, NotOnOrAfter, and <saml2:Audience></p>	<p>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.</p>	<pre><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></pre>

Element	Description	Example
<code><saml2:AttributeStatement></code> and <code><saml2:Attribute></code>	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".	<pre> <saml2:AttributeStatement > <saml2:Attribute Name= "external_user" > <saml2:AttributeValue xsi:type="xs:string">true </saml2:AttributeValue> </saml2:Attribute> </ saml2:AttributeStatement> </pre>

Related Information

[Deprecation of OAuth IdP API /oauth/idp](#)

3.2.1.3 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.

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	<code>https://<API-Server>/oauth/token</code>
Authentication	No Auth
Headers	<code>Content-Type: application/x-www-form-urlencoded</code>
Request Body	<p>Enter the following values in the format of x-www.-form-urlencoded:</p> <ul style="list-style-type: none">• <code>company_id</code>: Required. Your company ID.• <code>client_id</code>: Required. API key generated in Registering Your OAuth2 Client Application [page 20].• <code>grant_type</code>: Required. Set the value to "urn:ietf:params:oauth:grant-type:saml2-bearer".• <code>assertion</code>: Required. Enter the Base64-encoded assertion obtained from Generating a SAML Assertion [page 26].• <code>new_token</code>: 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 <code>expire_in</code> field. You can use parameter <code>new_token=true</code> to force the server to generate a new access token valid for 24 hours.

Sample response:

```
{
  "access_token": "eyJ0b2t1bknvbnRlbnQ***ZMm5Tdz0ifQ==",
  "token_type": "Bearer",
  "expires_in": 86399
}
```

Related Information

[IP Restrictions](#)

3.2.1.4 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.

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	<code>https://<API-Server>/oauth/validate</code>
Headers	<code>Authorization: Bearer <Your access token></code>

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
}
```

Related Information

[IP Restrictions](#)

3.3 Restricting API Access by IP Addresses or IP Address Ranges

You can use Admin Center tools to restrict users' access to APIs based on IP allowlists.

Prerequisites

You have the following permissions:

- [Adminstrator](#) > [Manage System Properties](#) > [Password & Login Policy Settings](#) >
- [Administrator](#) > [Manage System Properties](#) > [IP Restriction Management](#) >

Context

Access to SAP SuccessFactors HCM suite APIs can be restricted by IP on the instance level and on the user level. There are several Admin Center settings you can use to achieve this:

- The [IP Restriction Management](#) tool allows you to set IP restriction on the instance-level for user login and API access. In the case of API access, this setting is overridden by the [Set API login exceptions](#) setting.
- The [Set API login exceptions](#) page under [Password & Login Policy Settings](#) allows you to set IP restriction on individual users. This setting applies to SF APIs, OData APIs, REST APIs, and OAuth Authentication APIs (`/oauth/token` and `/oauth/validate`). It overrides the [IP Restriction Management](#) setting for API access. If not set, the system falls back on [IP Restriction Management](#) to restrict API access.
- Additionally, you can restrict users' access to OData APIs using Basic Authentication. For more information, see the Related Information section.

Follow the instructions below to set IP restrictions for API access to your systems.

Procedure

- To set IP restrictions on the instance level, go to [Admin Center](#) > [IP Restriction Management](#) >, and add a single IP address or IP address range from where users are allowed access to your system, including API access.

- To set IP restriction by users, go to ► [Admin Center](#) ► [Password & Login Policy Settings](#) ► and choose [Set API login exceptions...](#). Here, you can restrict API access by users based on a single IP address or IP address range. Complete the following information:

Option	Description
Username	Enter the username for whom you want to apply the policy.
Maximum password age (days)	Enter the number of days the password is valid for. To set a password to never expire, enter -1 . <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>For security reasons, we do not recommend setting a password to never expire. This setting has no effect on API access using OAuth.</p> </div>
IP address restrictions	Enter the IP addresses separated by comma. For example: 10.20.30.40, 10.20.30.1-10.20.30.10 . <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>The maximum password age applies only when the user is accessing from the IP addresses specified here.</p> <p>Only IPv4 addresses are supported.</p> </div>

Related Information

[Restricting OData API Access through Basic Authentication](#)

3.4 Integration Tools

This section provides an understanding of tools that are used to monitor traffic and get detailed information about API requests.

There are three tools that can be used to monitor traffic and/or get detailed payload information about API requests made to your system and to understand the SF data model.

API Audit Log

⚠ Caution

As of 2019 Q2 Release, the [SFAPI audit log setting](#) page has been removed from Provisioning as part of the SFAPI deprecation process. You're no longer able to set the maximum log count and maximum message body

count for SFAPI in Provisioning. Previous settings will remain effective. You can still enable and disable SFAPI audit log in API Center. Please note that SFAPI has been deprecated as of August 1, 2018. We encourage you to use OData for new development and move your legacy applications and integrations to OData.

The API Audit log captures payload details for the last 10,000 API calls. This log allows you to inspect exactly the API payload request made to the system and the corresponding API response sent by the system. The API Audit Log is intended to help with support and debugging of API usage. The intended end user will be a developer who is using the API during an implementation or an administrator who can share information in this log with Product Support to help resolve API related support issues. The tool allows you to download data from individual calls, which you can send to your SAP SuccessFactors Product Support representative.

Note

Since this is sensitive data, care should be taken when granting permission on who can see this audit log. One safe strategy would be to grant this feature to developers only when necessary for development and debugging purposes, and then remove access to this feature. The log will always be captured, regardless of whether a particular user may see the log or not. Therefore, removing a user's access to the log will not remove the actual data in the log.

API Metering Details

The API Metering Details page gives you analytics on API usage for the last 30 days. You can use this page to see API call history analytics like how many times the API was called, or what was the total record counts accessed in your system.

API Data Dictionary

The API Data Dictionary page will give you access to a list of all data entities in your SF instance. The Data Dictionary displays the field meta data for each entity in a convenient, readable format.

Permission Controls on Integration Tools

Permission to access any of the Integration Tools pages must be granted for each individual tool by an administrator of your SAP SuccessFactors system. Granting access to the Integration Tools pages depends on which permission system you are using in your SAP SuccessFactors instance. The following are instructions for each permission system.

User Based Permission System

Permission to access the Integration Tools is granted by an administrator of your SAP SuccessFactors instance. These permissions can be found in the "Administrative Privileges" page. This location of this page depends on which Admin Tools version you are using:

- Accessing Administrative Privileges page in new admin tools: The Administrative Privileges page is accessed through the Administrative Privileges link found under the Set User Permissions icon.

- Accessing Administrative Privileges page in new admin tools: If you are using the Old Admin Tools page, the Administrative Privileges page is accessed through the ► [Manage Security](#) ► [Administrative Privileges](#) ► link.
- Once you are on the Administrative Privileges page, you can set permission to access the Integration Tools through the following steps:
 1. Search for a user and select the checkbox for the user.
 2. Select the Integration Tools link.
 3. Select the checkbox to access the desired Integration Tools.
 4. Select the button Save Admin Permission for Selected Users.

Role-Based Permission System

In the Role Based Permission system, the permission to Access to Integration Tools is in the Manage Integration Tools section of the Permissions settings page.

Accessing the Integration Tools

The Integration Tools that a user has been granted permission to access will be visible as links under Integration Tools in the older version of Admin page. The links can be accessed from here in the new Admin Page.

3.5 API Thresholds and Limitations

A table outlining the limitations in the SFAPI.

There are several thresholds and limitations in the SFAPI.

Threshold / Limitation	Limit	Description
Maximum rows to process in a single Insert/Update/Upsert/Delete operation	200	By default, a maximum of 200 rows will be processed in a single Insert/Update/Upsert/Delete method. This number can be set to a value from 1 to 800 by specifying the batchSize parameter for insert, update, upsert or delete operations, or in the login method, which sets the default batchSize for the session.
Maximum rows to return in a single Query/QueryMore method	200	By default, a maximum of 200 rows will be returned in a single Query/QueryMore method. This number can be set to a value from 1 to 800 by specifying the maxRows parameter in the query method.

Threshold / Limitation	Limit	Description
Maximum message size (HTTP content size)	5 MB	A maximum SOAP message size (HTTP content-size) cannot exceed 5MB. This is the limit when upload binary attachment using SFAPI. In general, the total attachment storage size is controlled by the attachment storage configuration of the company instance
Number of adhoc reports per session	5	You can submit only five adhoc reports per session. The limit of the number of upsert calls you can make to BizX is set only by server capacity. We suggest using separate sessions to handle parallel API calls.
SFAPI audit log size	2 MB	The maximum audit log size for SFAPI response payload is 2 MB. Responses larger than 2 MB are not logged. This limit helps improve server performance and reduce out-of-memory errors.

4 SFAPI Operations

4.1 API Summary

API Summary

The SFAPI is intended to address import and export of SAP SuccessFactors data. It is designed in a generic approach. The data that you can access from the API is not defined in the WSDL. Instead you can use the API metadata operations to discover the data and its schema in your SAP SuccessFactors instance.

There are five categories of methods in the API:

- Session Management (login/logout/isValidSession)
- MetaData Inspection (list/describe)
- Data Manipulation (insert/update/upsert/delete)
- Data Query (query/queryMore)
- Async Job (submitJob/getJobStatus/getJobResult/listJobs/cancelJob)

Each of the operations above are documented in detail in the following sections.

Session Management Operations

LoginResult login(SFCredential credential, List<SFParameters> params)	Creates the SFAPI session
boolean logout()	Destroys the current API session
boolean isValidSession()	Returns true if the current API session is still valid

MetaData Operations

String[] list()	List the entities in your company instance
DescribeResult describe(String type[], SFParameter params[])	Returns metadata about the list of entities specified in the type parameter (both field and entity information)
DescribeExResult describeEx(String type[], SFParameter params[])	This method is in beta development and and is not stable yet. It will stabilize in the 1109 release. It will be an extended version of the describe method.

Data Manipulation Operations

The four data manipulation operations are equivalent to the SQL operations of insert, update, and delete, plus a combination of insert and update.

InsertResult insert(String type, SFOBJECT[] objects, SFPARAMETER[] processingParam)	Inserts the objects of the specified entity type. The operation will resume if one row has failed to insert.
UpdateResult update(String type, SFOBJECT[] objects, SFPARAMETER[] processinParam)	Updates the objects of the specified entity type. The operation will resume if one row has failed to update.
UpsertResult upsert(String type, SFOBJECT[] objects, SFPARAMETER[] processingParam)	Inserts or updates the objects of the specified entity type. If the row doesn't exist, perform the insert operation, if the row exists, perform the update operation. The operation will resume if one row has failed to upsert.
DeleteResult delete(String type, SFOBJECT[] objects, SFPARAMETER[] processingParam)	Deletes the SFOBJECTS specified by the type and the objects. The operation will resume if one row has failed to delete.

Data Query Operations

QueryResult query(String queryString, SFPARAMETER[] param)	Queries SAP SuccessFactors with the given query string in SFQL (SuccessFactors Query Language). For a detailed grammar description, please check the related section in this reference guide.
QueryResult queryMore(String querySessionId)	The queryMore call is provided to support paging. It requires a querySessionId to identify the next page of results. The querySessionId is a parameter of the QueryResult object, obtained from a previous call to either the query or queryMore operation.

Async Job Operations

TaskStatus submitQueryJob(String queryString, SFPARAMETER[] param)	Submit the asynchronous query job to SAP SuccessFactors with the given query string in SFQL (SuccessFactors Query Language). TaskStatus includes "taskId" which is used to identify a submitted job. For a detailed grammar description, please check the related section in this reference guide.
TaskStatus getJobStatus(String taskId)	Get the execution status of the submitted asynchronous job.
DataHandler getJobResult(GetJobResult parameters)	Download the result of submitted asynchronous query job.
TaskStatus[] listJobs()	List all Jobs which are in running or waiting to run.
TaskStatus cancelJob(String taskId)	Download the result of submitted asynchronous query job.

4.2 Session Management Operations

Before accessing any SAP SuccessFactors data, you must establish a login session. You can programmatically check to validate that you are in a login session and logout at the end of the session to maximize application security. Session management operations include:

API Signature Description

Signature	Description
Boolean isValidSession()	Validate that the session is still operational.
LoginResult login(SFCredential)	Create an API login session.
LogoutResult logout()	Destroy the login session.

API Framework Objects

The following objects are used in the session management operations.

LoginResult

Extended from the SFObjcet.

Name	Description
error	List of the error messages which occurred in the login process.
msUntilPwdExpiration	The number of the ms (milliseconds) until the password will expire.
sessionId	The sessionId of the created session.

SFCredential

Name	Description
companyId	Identifies the SAP SuccessFactors client instance.
developerKey	NOT USED. Leave it blank.
password	The password associated with the username.
username	Username for a valid user in the specified client instance. The user should have API Login Permission, and permissions to access any desired data. See more details in the Security Overview section in the chapter on Using SFAPI.

SFParameter

SFParameter is a generic name/value container object, with the following fields:

Name	Description
Name	The parameter name.
Value	The parameter value.

Login

LoginResult login(SFCredential credential, List<SFParameters> params)	Creates the SFAPI session. A successful API session is required for any subsequent API calls.
--	---

Supported Parameters

Name	Type	Description
credential	SFCredential	Specifies user login credentials.

Name	Type	Description
params	List<SFParameters>	batchSize will specify the default page-Size to use for data manipulation operations (Insert, Update or Upsert). This can be a value from 1 to 800. The system will default to pageSize of 200 if not specified.

Optional SFParameter Values For Login

Below is a table of SFParameter values that can be used in the Login operation to specify session settings.

SF Parameter	Value	Description
batchSize	An integer from 1 to 800	batchSize will specify the default page-Size to use for data manipulation operations (Insert, Update or Upsert). This can be a value from 1 to 800. The system will default to pageSize of 200 if not specified.

Example Java Code

```

Declaration
/** The WS service object. */
private SFAPIService fService;
/** The WS invocation proxy. */
private SFAPI fProxy;
Initialization
fService = new SFAPIService(getWSEndpointURL(endpointURL), new
QName(NS_SFAPI_SERVER, SFAPIService));
fProxy = fService.getSFAPI();
Map<String, Object> requestContext = ((BindingProvider) fProxy).getRequestContext();
// Need to set the property to maintain the http session.
requestContext.put(BindingProvider.SESSION_MAINTAIN_PROPERTY,true);
login()
SFCredential credential = new SFCredential(); credential.setCompanyId("acme");
credential.setUsername("user1"); credential.setPassword("pwd");
List<SFParameter> params = new LinkedList<SFParameter>(); LoginResult result
= fProxy.login(credential, params); List<com.successfactors.sfapi.sfobject.Error>
errorList = result.getError();
if (errorList != null && errorList.size() > 0) {
// Login failed
System.out.println("Failed to login!");
System.out.println("Error code : " + errorList.get(0).getErrorCode()
+ ". Error message: " + errorList.get(0).getErrorMessage());
return false;
}
System.out.println("Successfully logged into SAP SuccessFactors.");

```

Example SOAP Request XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
<soapenv:Header/>
<soapenv:Body>
<urn:login>
<urn:credential>
<urn:companyId>acme</urn:companyId>
<urn:username>user1</urn:username>
<urn:password>pwd</urn:password>
</urn:credential>
<!--Zero or more repetitions:-->
<urn:param>
<urn:name>batchSize</urn:name>
<urn:value>500</urn:value>
</urn:param>
</urn:login>
</soapenv:Body>
</soapenv:Envelope>
```

Example SOAP Response XML

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<loginResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
<result>
<sessionId>
676D4E614BC41AB6660A4DC8AE267F31
</sessionId>
<msUntilPwdExpiration>
9223372036854775807
</msUntilPwdExpiration>
(Expired time in ms, this value means never expires)
</result>
</loginResponse>
</S:Body>
</S:Envelope>
```

Example SOAP Failed Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<loginResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
<result>
<sessionId xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
<msUntilPwdExpiration xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"/>
```

```
<error>
<errorCode>FAILED_AUTHENTICATION</errorCode>
<errorMessage>
Login failure due to the invalid company!
</errorMessage>
</error>
</result>
</loginResponse>
</S:Body>
</S:Envelope>
```

LogOut

boolean logout()

Destroys the current API session.

Supported Parameters

None.

Example Java Code

```
if (fProxy.logout()) { System.out.println("Logout Successfully.");
...
} else {
System.out.println("Failed to logout");
}
```

Example SOAP Request XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
<soapenv:Header/>
<soapenv:Body>
<urn:logout/>
</soapenv:Body>
</soapenv:Envelope>
```

Example SOAP Response XML

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
```

```
<logoutResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
<result>>true</result>
</logoutResponse>
</S:Body>
</S:Envelope>
```

Example SOAP Failed Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<logoutResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
<result>>false</result>
</logoutResponse>
</S:Body>
</S:Envelope>
```

4.3 MetaData Discovery Methods

The MetaData methods allow for dynamic discovery of SFObjets available in the API. This allows you to list all the objects in the system, and describe objects to inspect their structure and provide meta information about the object, like which API data manipulation operations are supported for this object.

API Signature

Signature	Description
DescribeResult[] describe(String[] types, List<SFParameter> params)	Describes the SFObjets specified by the list of entity names. Note you can get all the available entity names of the SFObjets using the list() call below. The returned DescribeResult contains metadata to describe the object properties in terms of name, data type and other important metadata.
List<String> list()	Retrieves a list the entity names of all SFObjets available through the SFAPI. You can use these names in the describe(String[] types) call.

API Framework Objects

DescribeResult

DescribeResult has the following properties:

Name	Description
error	List of error codes and messages.
feature	The supported feature list, valid values are: insert, upsert, delete, update, query, queryMore.
field	An array of fields found in the SFObjct. The Field object contains metadata about the field like field name, data type, etc. See the Field object definition for more information.
type	Describes the type of this SFObjct. For example, "User" or "Goal\$1". The type is also referred to as the entity name. This value will correspond to one of the values returned in the API list() operation.

Field Object

The Field object contains metadata about an SFObjct field. It is necessary for runtime discovery of field properties. The Field object has the following properties:

Name	Description
dataType	Specifies the field data type. One of the following: Integer Long Float Double String Boolean Date (with date only) DateTime (with both date and time) Binary

Name	Description
label	A list of localized labels for this field.
maxlength	If not null, this field specifies the maximum length restriction (byte count in UTF-8 format) on the field. Any data beyond this length will be truncated.
name	A unique name for the field.
picklist	A Picklist object which provides enumerated values for this field. These can be configured / set up in the Administrative area of the SAP SuccessFactors system.
required	Flag to indicate if this is a required field.

Field Object Data Types

The following data types are supported for fields of SFObjets in the SFAPI:

Type Name	Description
The type for binary data.	Embedded byte stream encoded in Base64 format.
Boolean type	The accepted value is "true" or "false"(ignore case).
Date Type(date only)	The accepted format is YYYY-MM-DD. For ex, 2008-02-23.
Date and Time Type(date and time)	The accepted format is in ISO 8601 format in UTC (YYYY-MM-DDThh:mm:ssZ). For example, 1999-01-01T23:01:01Z (In time zone GMT).
Double type(Signed)	8 bytes double. The maximum value is 1.7976931348623157e+308.
Float type(Signed)	4 bytes float. The maximum value is 3.4028235e+38.
Integer type(Signed)	4 bytes integer. The range is (-2 ³¹ ~ 2 ³¹ -1).
Long type(Signed)	8 bytes long. The range is (-2 ⁶³ ~ 2 ⁶³ -1).
String type	The maxlength field will return the maximum allowed bytes when the string is converted to UTF-8 encoding.

Picklist

Picklist has the following properties:

Name	Type	Description
id	String	A unique id for this picklist.
picklistOptions	PicklistOption []	An array of selectable options for this picklist.
source	String	The source is used.

Picklist Option

PicklistOption has the following properties:

Name	Type	Description
externalCode	String	An external code that can be associated with this value.
id	Integer	A unique id for this picklist option.
labels	Label []	A list of localized labels for this picklist option.
status	String	A status for this picklist. Status can be "ACTIVE", "OBSOLETE" or "DELETED". Only "ACTIVE" picklist options will be displayed in the SAP SuccessFactors application.
value	Integer	An integer value assigned to this picklist option.

Label

A localized label object. Label contains the following properties:

Name	Type	Description
locale	String	Optional. The locale for this label, where the locale = language code + “_” + country code + “_” + variant. The language code should adhere to ISO 639, the country code to ISO 3166. Variant has no requirements on format. The following are valid locale strings: “en” – English language “en_US” – English language specific to the US “en_US_CA” – English in California, U.S.A.
mime-type	String	Allows mime type specification of either “text-plain”, or “text-html”. If mime type of “text-html” is specified then the SAP SuccessFactors application will interpret the label as html, otherwise it will be rendered as plain text.
value	String	The localized label.

List

String[] list()	List the entities in your company instance.
-----------------	---

Supported Parameters

None.

Example Java Code

```
ListSFObjects param = new ListSFObjects();
ListSFObjectsResponse result = fProxy.list(param);
System.out.println("Entity list:");
for (String fieldName : result.getName()) {
```



```

System.out.println("\t" + fieldName);
}

```

Example SOAP Request XML

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
<soapenv:Header/>
<soapenv:Body>
<urn:listSFObjects/>
</soapenv:Body>
</soapenv:Envelope>

```

Example SOAP Response XML

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<listSFObjectsResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
<name>User</name>
<name>LearningActivity$4001</name>
<name>MatrixManager</name>
<name>CustomManager</name>
<name>SecondManager</name>
<name>Background_googledocs</name>
<name>Background_insideWorkExperience</name>
<name>Background_specialAssign</name>
<name>Background_outsideWorkExperience</name>
<name>Background_education</name>
<name>Background_courses</name>
<name>Background_certificates</name>
<name>Background_awards</name>
<name>Background_languages</name>
<name>Background_funcExperience</name>
<name>Background_leadExperience</name>
<name>Background_preferredNextMove</name>
<name>Background_mobility</name>
<name>Background_memberships</name>
<name>Background_community</name>
<name>Background_promotability</name>
<name>Background_compensation</name>
<name>Background_varPayEmpHistData</name>
<name>CalibrationSubject</name>
<name>JobPosting</name>
<name>BackgroundCheck</name>
<name>Onboarding</name>
<name>TrendData_sysOverallPerformance</name>
<name>TrendData_sysOverallPotential</name>
<name>TrendData_sysOverallObjective</name>
<name>TrendData_sysOverallCompetency</name>
<name>Goal$1</name>
<name>Goal$3</name>
</listSFObjectsResponse>
</S:Body>
</S:Envelope>

```

Describe

DescribeResult describe(String type[], SFParameter params[])

Returns metadata about the list of entities specified in the type parameter.

Supported Parameters

Name	Type	Description
params	List <SFParameters>	NOT USED. Leave it null.
type	String []	A list of string values that specify entity names (entity types) that you want to describe. These values come from the list() operation.

Example Java Code

```
DescribeSFObjects param = new DescribeSFObjects();
param.getType().add("User");
DescribeSFObjectsResponse response = fProxy.describe(param);
for (DescribeResult result : response.getResult()) {
    System.out.println("Entity Type : " + result.getType());
    int index = 0;
    for (Field field : result.getField()) {
        System.out.println("Field #" + index + " : " +
            field.getName()); System.out.println("\tRequired : " + field.isRequired());
        System.out.println("\tData Type : " + field.getDataType().value());
        System.out.println("\tMax Length : " + field.getMaxlength()); List<Label> labelList
            = field.getLabel();
        if (labelList != null & labelList.size() > 0) {
            for (Label label : field.getLabel()) {
                System.out.println("\tLabel : "); System.out.println("\t Value : " +
                    label.getValue());
                System.out.println("\t Locale : " + label.getValue());
            }
        }
        Picklist picklist = field.getPicklist();
        if (picklist != null) { System.out.println("\tPick List: ");
            System.out.println("\t Id : " + picklist.getId());
            System.out.println("\t Source : " + picklist.getSource());
            ++index;
        }
        System.out.println("Supported Operations : ");
        for (FeatureType featureType : result.getFeatureType()) {
            System.out.println(" " + featureType.value());
        }
    }
}
```

Example SOAP Request XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:describeSFObjects>
      <!--1 or more repetitions-->
      <urn:type>LearningActivity$4001</urn:type>
      <!--1 or more repetitions-->
      <urn:param>
        <urn:name>locale</urn:name>
        <urn:value>en-US</urn:value>
      </urn:param>
    </urn:describeSFObjects>
  </soapenv:Body>
</soapenv:Envelope>
```

Example SOAP Response XML

```
<S:Body>
  <describeSFObjectsResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
    <result>
      <type>LearningActivity$4001</type>
      <field>
        <name>GUID</name>
        <dataType>string</dataType>
        <maxlength>256</maxlength>
        <required>true</required>
      </field>
      <field>
        <name>ParentId</name>
        <dataType>string</dataType>
        <maxlength>100</maxlength>
        <required>false</required>
      </field>
      <!-- more fields -->
      <feature>upsert</feature>
    </result>
  </describeSFObjectsResponse>
</S:Body>
```

Example SOAP Failed Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:Fault xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns3="http://
www.w3.org/2003/05/soap-envelope">
      <faultcode>ns2:Server</faultcode>
      <faultstring>SFAPI Domain Error!</faultstring>
      <detail>
        <ns2:SFWebServiceFault xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
          <ns2:errorCode>
```

```

UNDEFINED_ENTITY_ID
</ns2:errorCode>
<ns2:errorMessage>
Entity type 'Learn22ingActivity$4001' is undefined!
</ns2:errorMessage>
</ns2:SFWebServiceFault>
</detail>
</ns2:Fault>
</S:Body>
</S:Envelope>

```

Note

Locale only affect the returned localized labels for each field.

DescribeEx

DescribeExResult describeEx(String type[], SFParameter params[])

This method is in beta development and will change in the 1109 release. It will be an extended version of the describe method.

Supported Parameters

Name	Type	Description
params	List <SFParameters>	NOT USED. Leave it null.
type	String []	A list of string values that specify entity names (entity types) that you want to describe. These values come from the list() operation.

Example SOAP Request XML

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
<soapenv:Header/>
<soapenv:Body>
<urn:describeSFObjectsEx>
<!--1 or more repetitions:-->
<urn:type>LearningActivity$4001</urn:type>
<!--1 or more repetitions:-->
<urn:param>
<urn:name>locale</urn:name>
<urn:value>en-US</urn:value>
</urn:param>
</urn:describeSFObjectsEx>
</soapenv:Body>

```

```
</soapenv:Envelope>
```

Example SOAP Response XML

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <describeSFObjectsExResponse xmlns="urn:sfobject.sfapi.successfactors.com"
      xmlns:ns2="urn:fault.sfapi.successfactors.com">
      <result>
        <type>learningactivity$4001</type>
        <field>
          <name>GUID</name>
          <dataType>string</dataType>
          <maxlength>256</maxlength>
          <required>true</required>
          <insertable>false</insertable>
          <upsertable>true</upsertable>
          <supportInOperator>false</supportInOperator>
          <updateable>false</updateable>
          <selectable>false</selectable>
          <filterable>false</filterable>
          <supportLikeOperator>false</supportLikeOperator>
          <sortable>false</sortable>
        </field>
        <field>
          <name>ParentId</name>
          <dataType>string</dataType>
          <maxlength>100</maxlength>
          <required>false</required>
          <insertable>false</insertable>
          <upsertable>true</upsertable>
          <supportInOperator>false</supportInOperator>
          <updateable>false</updateable>
          <selectable>false</selectable>
          <filterable>false</filterable>
          <supportLikeOperator>false</supportLikeOperator>
          <sortable>false</sortable>
        </field>
        <!-- more fields -->
        <feature>upsert</feature>
      </result>
    </describeSFObjectsExResponse>
  </S:Body>
</S:Envelope>
```

Example SOAP Failed Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:Fault xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns3="http://
      www.w3.org/2003/05/soap-envelope">
      <faultcode>ns2:Server</faultcode>
      <faultstring>SFAPI Domain Error!</faultstring>
      <detail>
        <ns2:SFWebServiceFault xmlns="urn:sfobject.sfapi.successfactors.com"
          xmlns:ns2="urn:fault.sfapi.successfactors.com">
          <ns2:errorCode>UNDEFINED_ENTITY_ID</ns2:errorCode>
        </ns2:SFWebServiceFault>
      </detail>
    </ns2:Fault>
  </S:Body>
</S:Envelope>
```

```

<ns2:errorMessage>
Entity type 'Learn22ingActivity$4001' is undefined!
</ns2:errorMessage>
</ns2:SFWebServiceFault>
</detail>
</ns2:Fault>
</S:Body>
</S:Envelope>

```

Note

Locale specifies the returned localized labels for each field.

4.4 Data Manipulation Operations

The data manipulation operations provide the create, read, update, and delete operations on the SFObjets exposed in the API. Refer to the separate SFAPI Data Dictionary document to ensure you have the proper permissions setup to access the desired SFObjets.

There are four data manipulation operations, insert/update/upsert and delete().

API Signature

Signature	Description
DeleteResult[] delete(String type, List<SFObjct> sfojct, List <SFParameter> processingParam)	Delete the SFObjets specified by the id field in the SFObjct list, and report an error if the id is not found.
InsertResult[] insert(String type, List<SFObjct> sfojct, List <SFParameter> processingParam)	Insert the list of SFObjets. Errors will be reported if the rows to be inserted already exist in the system. Consult the individual SFObjct types in the separate SFAPI Data Dictionary for any processingParam that may apply.
UpdateResult[] update(String type, List<SFObjct> sfojct, List <SFParameter> processingParam)	Update the rows specified by the id field in the SFObjct list, and report errors if the row id does not exist in the system. Consult the individual SFObjct types types in the separate SFAPI Data Dictionary for any processingParam that may apply.

Signature	Description
UpsertResult[] upsert(String type, List<SFOBJECT> sfoobject, List <SFParameter> processingParam)	<p>The upsert operation will insert or update the given list of SFOBJECTS. This operation will check if the specified row exists and then will update the row, and if the row is not found, insert the row.</p> <p>Consult the individual SFOBJECT types in the separate SF-API Data Dictionary for any processingParam that may apply.</p>

API Framework Objects

SFOBJECT

SFOBJECT is a generic class for all SF-API data objects. An SFOBJECT is analogous to a row in a database table. Each SFOBJECT will have fields that are evaluated at runtime. As a generic object the fields are implemented as un-typed data in the generic xsd<any> element.

The allowable fields for the SFOBJECT can be discovered using the describe operation. Below are the attributes of the SFOBJECT. There are only three, the ID, the type and the xsd <any> element which will contain the data fields requested in the queryString, or supplied by the user for import type operations.

SFOBJECT has the following fields:

Name	Type	Description
any	List<Object>	<p>The list of the SFOBJECT fields. The format is:</p> <p><FieldName>FieldValue</FieldName>.</p> <p>The list element is a DOM object.</p>
id	String	<p>A unique identifier for the object. This is required field when in update/delete operations. In insert/upsert operations, this field MUST NOT exist.</p>
type	String	<p>The type of this object.</p> <p>For example "User" or "Transcript".</p>

SFParameter

SFParameter is a generic structure used to pass in name - value pairs of information to control method processing. For example, in user import you can pass in a parameter to control whether welcome messages are sent to new users.

SFParameter has the following properties:

Property Name	Type	Description
Name	String	The parameter name.
Value	String	The parameter value.

InsertResult/UpdateResult/DeleteResult/UpsertResult

All four of these result type objects have the same format. The result objects are used to contain results and status of the data manipulation request - where multiple objects were processed during the request. In addition to the overall status and error code, the status of the each row processed has a specific status and error message.

Name	Type	Description
jobStatus	String	<p>Indicates if there were errors encountered while processing the operations.</p> <p>Values can be one of the following:</p> <p>OK</p> <p>ERROR</p> <p>If all the rows are validated and processed without error, the value is OK.</p> <p>Check the row level detail status/error if the return value is ERROR.</p> <p>For ex, if the required externalId is not specified in insert operation, the status will return ERROR code. Check the row level error by examining the objectEditResult for which row is causing the error.</p>
Message	String	The message from this job. Usually this is a summary field indicating the number of rows processed, the number of rows failed and the processing time.

Name	Type	Description
objectEditResult	List<ObjectEditResult>	An array of the edit results for the objects that were imported. Each row in the request message will have a related ObjectEditResult.

ObjectEditResult

ObjectEditResult is used to contain the results and status of edit related calls on an object.

ObjectEditResult has the following properties:

Name	Type	Description
errorStatus	String	Indicates if there was an error editing the object. Values can be one of the following: OK ERROR NO_MATCH (only in Update or Delete operations) Note that even if there was an ERROR encountered, the editStatus below should still be checked. An ERROR does not mean the editStatus is automatically NO_EDIT.
Index	Integer	A zero based index into the input array of SFObjets that was submitted for edits to which this ObjectEditResult applies.
editStatus	String	Indicates what type of edit operation was performed on the object. Values can be one of the following: CREATED UPDATED DELETED NO_EDIT
message	String	Contains an error message if an error was encountered. (i.e., the errorStatus = ERROR).

Insert

InsertResult insert(String type, SFOBJECT[] objects, SFPARAMETER[] processingParam)

Insert the list of SFOBJECTS. Errors will be reported if the rows already exist in the system.

Supported Parameters

Parameter Name	Type	Description
objects	SFOBJECT []	A list of SFOBJECTS that you want to insert. Errors will be reported if the rows already exist in the system, or if you are missing any required fields. The required fields can be discovered using the describe() operation for this entity.
processingParam	List <SFPARAMETERS>	List of optional parameters.
type	String	The entity type name that you want to insert. This must correspond to a value returned by the list() operation.

Example Java Code

```
List<SFOBJECT> sfojects = new LinkedList<SFOBJECT>();
// For upsert / insert operation, all required fields are needed. Please see
// the describe result to find out which fields are required.
// Fill user info for user1
SFOBJECT user1 = new SFOBJECT();
user1.setType("User");
SOAPFACTORY factory = SOAPFACTORY.newInstance();
SOAPELEMENT element = factory.createElement("externalId");
element.addTextNode("user_1");
user1.getAny().add(element);
element = factory.createElement("username");
element.addTextNode("user1");
user1.getAny().add(element);
element = factory.createElement("firstName");
element.addTextNode("Carla");
user1.getAny().add(element);
element = factory.createElement("lastName");
element.addTextNode("Grant");
user1.getAny().add(element);
element = factory.createElement("email");
element.addTextNode("cgrant@bestrun.com");
user1.getAny().add(element);
sfojects.add(user1);
InsertResult result = fProxy.insert("User", sfojects, null);
System.out.println("Insert status : " + result.getJobStatus());
System.out.println("Message : " + result.getMessage());
for (OBJECTEDITRESULT objectEditResult : result.getObjectEditResult())
```

```

{
System.out.println("Object #" + objectEditResult.getIndex());
System.out.println("\tId : " + objectEditResult.getId());
System.out.println("\tEdit Status : " + objectEditResult.getEditStatus());
System.out.println("\tError Status : " + objectEditResult.getErrorStatus());
}

```

Example SOAP Request XML

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
<soapenv:Header/>
<soapenv:Body>
<urn:insert>
<urn:type>user</urn:type>
<!--1 or more repetitions:-->
<urn:sfobject>
<urn:type>user</urn:type>
<externalId>cgrant</externalId>
<username>cgrant</username>
<status>active</status>
<password>pwd</password>
<firstName>Carla</firstName>
<lastName>Grant</lastName>
<email>cgrant@bestrun.com</email>
<managerId>USR-8</managerId>
<hrId>USR-8</hrId>
<department>test</department>
<jobCode>test</jobCode>
<division>test</division>
<location>test</location>
<timeZone>PST</timeZone>
<hireDate>2010-01-01</hireDate>
</urn:sfobject>
</urn:insert>
</soapenv:Body>
</soapenv:Envelope>

```

Example SOAP Response XML

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<insertResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
<result>
<jobStatus>OK</jobStatus>
<message xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
<objectEditResult>
<id>USR-4389</id>
<errorStatus>OK</errorStatus>
<editStatus>CREATED</editStatus>
<index>0</index>
<message xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
</objectEditResult>
</result>
</insertResponse>
</S:Body>

```

```
</S:Envelope>
```

Example SOAP Failed Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <insertResponse xmlns="urn:sfobject.sfapi.successfactors.com"
      xmlns:ns2="urn:fault.sfapi.successfactors.com">
      <result>
        <jobStatus>ERROR</jobStatus>
        <message xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
        <objectEditResult>
          <errorStatus>ERROR</errorStatus>
          <editStatus>NOEDIT</editStatus>
          <index>0</index>
          <message>INTERNAL_ERROR : User already exists.</message>
        </objectEditResult>
      </result>
    </insertResponse>
  </S:Body>
</S:Envelope>
```

Note

Insert and Upsert cannot have "ID" field in the SFOBJECT

Update

UpdateResult update(String type, SFOBJECT[] objects, SFParameter[] processingParam)

Updates the objects of the specified entity type. The operation will resume if one row has failed to update.

Supported Parameters

Parameter Name	Type	Description
objects	SFOBJECT []	A list of SFOBJECTS that you want to update. Errors will be reported if the rows do not exist in the system.
processingParam	List <SFParameters>	List of optional parameters.
type	String	The entity type name that you want to update. This must correspond to a value returned by the list() operation.

Example SOAP Request XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:update>
      <urn:type>user</urn:type>
      <!--1 or more repetitions-->
      <urn:sfobject>
        <urn:id>USR-4389</urn:id>
        <urn:type>user</urn:type>
        <firstName>test</firstName>
        <password>123</password>
      </urn:sfobject>
    </urn:update>
  </soapenv:Body>
</soapenv:Envelope>
```

Example SOAP Response XML

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <updateResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
      <result>
        <jobStatus>OK</jobStatus>
        <message xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
        <objectEditResult>
          <id>USR-4389</id>
          <errorStatus>OK</errorStatus>
          <editStatus>UPDATED</editStatus>
          <index>0</index>
        </objectEditResult>
        <message xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
      </result>
    </updateResponse>
  </S:Body>
</S:Envelope>
```

Example SOAP Failed Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <updateResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
      <result>
        <jobStatus>OK</jobStatus>
        <message xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
        <objectEditResult>
          <errorStatus>NO_MATCH</errorStatus>
          <editStatus>NOEDIT</editStatus>
          <index>0</index>
          <message>
            USER_NOT_FOUND : Cannot find user by user id. InternalId = USR-224389
          </message>
        </objectEditResult>
      </result>
    </updateResponse>
  </S:Body>
</S:Envelope>
```

```

</message>
</objectEditResult>
</result>
</updateResponse>
</S:Body>
</S:Envelope>

```

Note

Update requires the ID key. Note you can update password just like other normal fields. For security, the password content will not be saved in the log or audit history.

Upsert

UpsertResult upsert(String type, SFObject[] objects, SFParameter[] processingParam)

Inserts or updates the objects of the specified entity type. If the row doesn't exist, perform the insert operation, if the row exists, perform the update operation. The operation will resume if one row has failed to upsert.

Supported Parameters

Parameter Name	Type	Description
objects	SFObject []	A list of SFObjects that you want to upsert. Errors will be reported if you are missing any required fields. The required fields can be discovered using the describe() operation for this entity
processingParam	List <SFParameters>	List of optional parameters.
type	String	The entity type name that you want to upsert. This must correspond to a value returned by the list() operation.

Example Java Code

```

List<SFObject> sfobjects = new LinkedList<SFObject>();
// For upsert / insert operation, all required fields are needed. Please see
// the describe result to find out which fields are required.
// Fill user info for user1
SFObject user1 = new SFObject();
user1.setType("User");
SOAPFactory factory = SOAPFactory.newInstance();
SOAPElement element = factory.createElement("externalId");
element.addTextNode("user_1");

```

```

user1.getAny().add(element);
element = factory.createElement("username");
element.addTextNode("user1"); user1.getAny().add(element);
element = factory.createElement("firstName");
element.addTextNode("Carla");
user1.getAny().add(element);
element = factory.createElement("lastName");
element.addTextNode("Grant");
user1.getAny().add(element);
element = factory.createElement("email");
element.addTextNode("cgrant@bestrun.com");
user1.getAny().add(element);
sfojects.add(user1);
UpsertResult result = fProxy.upsert("User", sfojects, null);
System.out.println("Upsert status : " + result.getJobStatus());
System.out.println("Message : " + result.getMessage());
for (ObjectEditResult objectEditResult : result.getObjectEditResult())
{
System.out.println("Object #" + objectEditResult.getIndex());
System.out.println("\tId : " + objectEditResult.getId());
System.out.println("\tEdit Status : " + objectEditResult.getEditStatus());
System.out.println("\tError Status : " + objectEditResult.getErrorStatus());
}

```

Example SOAP Request XML

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
<soapenv:Header/>
<soapenv:Body>
<urn:upsert>
<urn:type>user</urn:type>
<!--1 or more repetitions:-->
<urn:sfobject>
<urn:type>user</urn:type>
<externalId>cgrant</externalId>
<username>cgrant</username>
<status>active</status>
<password>cgrant</password>
<firstName>test</firstName>
<lastName>test</lastName>
<mi>test</mi>
<nickname>test</nickname>
<suffix>test</suffix>
<salutation>test</salutation>
<gender>M</gender>
<email>test@test.com</email>
<managerId>USR-8</managerId>
<hrId>USR-8</hrId>
<department>test</department>
<jobCode>test</jobCode>
<division>test</division>
<location>test</location>
<timeZone>PST</timeZone>
<hireDate>2010-01-01</hireDate>
<empId>test</empId>
<title>test</title>
<businessPhone>test</businessPhone>
<fax>test</fax>
<addressLine1>test</addressLine1>
<addressLine2>test</addressLine2>
<city>test</city>
<state>test</state>

```

```

<zipCode>test</zipCode>
<country>test</country>
<reviewFreq>test</reviewFreq>
<lastReviewDate>test</lastReviewDate>
<custom01>test</custom01>
<!-- more fields from the describe -->
</urn:sfobject>
</urn:upsert>
</soapenv:Body>
</soapenv:Envelope>

```

Example SOAP Response XML

```

<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<upsertResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
<result>
<jobStatus>OK</jobStatus>
<message xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
<objectEditResult>
<id>USR-4388</id>
<errorStatus>OK</errorStatus>
<editStatus>UPDATED</editStatus>
<index>0</index>
<message
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:nil="true"/>
</objectEditResult>
</result>
</upsertResponse>
</S:Body>
</S:Envelope>

```

Example SOAP Failed Response

```

<?xml version='1.0' encoding='UTF-8'?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<ns2:Fault xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns3="http://
www.w3.org/2003/05/soap-envelope">
<faultcode>ns2:Server</faultcode>
<faultstring>SFAPI Domain Error!</faultstring>
<detail>
<ns2:SFWebServiceFault xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
<ns2:errorCode>
INVALID_REQUEST_MESSAGE
</ns2:errorCode>
<ns2:errorMessage>
Invalid date request message! Error: Invalid date value 2010-012-01 at
message#=1,field#=108,field=servicedate!
</ns2:errorMessage>
</ns2:SFWebServiceFault>
</detail>
</ns2:Fault>
</S:Body>

```


</S:Envelope>

Note

Upsert should return success code of "UPDATED" or "CREATED", otherwise it will return operation failed error code and message.

Delete

Supported Parameters

Parameter Name	Type	Description
Objects	SFObject []	A list of SFObjects that you want to delete. Errors will be reported if the row does not exist.
processingParam	List <SFParameters>	List of optional parameters.
Type	String	The entity type name that you want to delete from. This must correspond to a value returned by the list() operation.

Example Java Code

```
List<SFObject> sfobjects = new LinkedList<SFObject>();
// Fill user info for user1
SFObject mmgr1 = new SFObject();
user1.setType("MatrixManager");
user1.setId("MMGR-1484");
sfobjects.add(mmgr1);
DeleteResult result = fProxy.delete("MatrixManager", sfobjects, null);
System.out.println("Update status : " + result.getJobStatus());
System.out.println("Message : " + result.getMessage());
for (ObjectEditResult objectEditResult : result.getObjectEditResult())
{
System.out.println("Object #" + objectEditResult.getIndex());
System.out.println("\tId : " + objectEditResult.getId());
System.out.println("\tEdit Status : " + objectEditResult.getEditStatus());
System.out.println("\tError Status : " + objectEditResult.getErrorStatus());
System.out.println("\tError Message : " + objectEditResult.getMessage());
}
```

Example SOAP Request XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:delete>
      <urn:type>MatrixManager</urn:type>
      <!--1 or more repetitions:-->
      <urn:sfobject>
        <!--Optional:-->
        <urn:id>MMGR-1484</urn:id>
        <urn:type>MatrixManager</urn:type>
      </urn:sfobject>
    </urn:delete>
  </soapenv:Body>
</soapenv:Envelope>
```

Example SOAP Response XML

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <deleteResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
      <result>
        <jobStatus>OK</jobStatus>
        <message/>
        <objectEditResult>
          <id>MMGR-1484</id>
          <errorStatus>OK</errorStatus>
          <editStatus>DELETED</editStatus>
          <index>0</index>
          <message xsi:nil="true" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
        </objectEditResult>
      </result>
    </deleteResponse>
  </S:Body>
</S:Envelope>
```

Example SOAP Failed Response

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns2:Fault xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns3="http://
www.w3.org/2003/05/soap-envelope">
      <faultcode>ns2:Server</faultcode>
      <faultstring>SFAPI Domain Error!</faultstring>
      <detail>
        <ns2:SFWebServiceFault xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
          <ns2:errorCode>UNSUPPORTED_OPERATION</ns2:errorCode>
          <ns2:errorMessage>
            Entity type 'User' doesn't support DELETE operation!
          </ns2:errorMessage>
        </ns2:SFWebServiceFault>
      </detail>
    </ns2:Fault>
  </S:Body>
</S:Envelope>
```

```
</detail>
</ns2:Fault>
</S:Body>
</S:Envelope>
```

4.5 Data Query Operations

The interfaces included data query feature using the SFQL. The methods also support query by pagination.

API Signature

Signature	Description
<code>QueryResult query(String queryString, List<SFParameter> params)</code>	<p>Queries the SAP SuccessFactors platform using SFQL query string. The maximum rows count is 200. Use <code>queryMore()</code> to retrieve the data page by page.</p> <p>The returned <code>QueryResult</code> contains the matching <code>SFObjects</code> specified by SFQL, the row count returned and the <code>querySessionId</code> used with subsequent <code>queryMore()</code> method call to retrieve the data.</p>
<code>QueryResult queryMore(String querySessionId)</code>	Retrieve the next page data of the previous query.

API Framework Objects

QueryResult

QueryResult has the following properties:

Name	Type	Description
hasMore	boolean	Flag to indicate if there are more result sets available. For example, the following code is the pattern <pre>r = query("select * from User"); while (r.hasMore()) { r = queryMore(r.querySessionId); }</pre>
numResults	int	Return the row count of the matching SFObjets.
querySessionId	String	A unique identifier used to pass into the queryMore operation if there are more rows. This will identify the next page of rows to retrieve in the queryMore operation.
sbject	SFObject[]	Returns the list of the SFObjects.

Query

QueryResult query(String queryString, SFParameter[] param) Queries the SAP SuccessFactors system with the given query string in SFQL (SuccessFactors Query Language). For a detailed grammar description, please check the related section in this reference guide.

The following is an example query String:

```
SELECT FirstName, LastName, JobCode, Title FROM User WHERE externalId = 'cgrant'.
```

The QueryResult object will contain the first page of results. Paging through the remaining results is accomplished using the queryMore() operation. The QueryResult object will return a boolean parameter "hasMorePages" to indicate if there are more pages remaining and a querySessionId, which must be passed into the queryMore() method to get the next page of results.

You can set the page size for a query by setting the "maxRows" parameter in the query method to an integer value from 1 to 800. The default page size is 200, and the maximum page size is 800.

Supported Parameters

startingRow, maxRows

Optional SFPParameter Values For Query

Below is a table of SFPParameter values that can be used in the query operation.

SFPParameter Name	SFPParameter Value	Description
maxRows	An integer from 1 to 800	maxRows will specify the pageSize for the returned result rows of the query. This can be a value from 1 to 800. The system will default to pageSize of 200 if not specified.

Example Java Code

```
query() & queryMore() query() & queryMore()
// display up to 25 records in all int numRowsToPrint = 25;
String sfql = "SELECT externalId, userName, firstName, lastName, email FROM User";
List<SFPParameter> paramList = new LinkedList<SFPParameter>();
SFPParameter param = new SFPParameter();
// Set query page size to retrieve 10 records per page param.setName("maxRows");
param.setValue("10");
paramList.add(param);
QueryResult result = fProxy.query(sfql, paramList);
int count = 0;
Utils.dumpQueryResult(result, numRowsToPrint, count);
count += result.getNumResults();
numRowsToPrint -= result.getNumResults();
while (numRowsToPrint > 0 && result.hasMore() && result.getNumResults() > 0) {
result = fProxy.queryMore(result.getQuerySessionId());
// Dump the first n limits rows Utils.dumpQueryResult(result, numRowsToPrint,
count); count += result.getNumResults();
numRowsToPrint -= result.getNumResults();
}
System.out.println("The total row count returned is " + count + ".");
/**
 * Dump the query result.
 *
 * @param result
 *     the query result.
 * @param numRowsToPrint
 *     the remaining rows to dump.
 * @param startRowIndex
 *     the start row index.
 */
public static void dumpQueryResult(QueryResult result, int numRowsToPrint, int
startRowIndex) {
System.out.println("The number of the records in a query: " +
result.getNumResults());
int index = 0;
```



```

</sfobject>
<sfobject>
<id>USR-4386</id>
<type>User</type>
<username>test_110124102601</username>
</sfobject>
<sfobject>
<id>USR-4387</id>
<type>User</type>
<username>test2_110124102602</username>
</sfobject>
<sfobject>
<id>USR-4388</id>
<type>User</type>
<username>cgrant</username>
</sfobject>
<sfobject>
<id>USR-4389</id>
<type>User</type>
<username>cgrant_2</username>
</sfobject>
<numResults>5</numResults>
<hasMore>>false</hasMore>
<querySessionId>
3209cbb1-2302-41ea-af76-b8158f34bd7d
</querySessionId>
</result>
</queryResponse>
</S:Body>
</S:Envelope>

```

Example SOAP Failed Response

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<ns2:Fault xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns3="http://
www.w3.org/2003/05/soap-envelope">
<faultcode>ns2:Server</faultcode>
<faultstring>SFAPI Domain Error!</faultstring>
<detail>
<ns2:SFWebServiceFault xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
<ns2:errorCode>UNDEFINED_ENTITY_ID</ns2:errorCode>
<ns2:errorMessage>
Entity type 'user23' is undefined!
</ns2:errorMessage>
</ns2:SFWebServiceFault>
</detail>
</ns2:Fault>
</S:Body>
</S:Envelope>

```

Note

Query Session returned in query is used for queryMore operation (pagination), but the server only keeps the last 5 query session ids. Older query operations expire. startingRow starts with 1, not 0.

QueryMore

QueryResult queryMore(String querySessionId)

The queryMore operation is used to support paging through results generated by the query operation.

Like the query operation, the queryMore operation also returns a QueryResult object, which contains the next page of results, and a hasMoreResults parameter to indicate if there are more results, along with a new querySessionId to be used in the next queryMore call.

Supported Parameters

None

Example Java Code

See the example listed for the query operation.

Example SOAP Request XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
<soapenv:Header/>
<soapenv:Body>
<urn:queryMore>
<urn:querySessionId>
4afa1b00-81f5-4d37-be72-382967c4cb22
</urn:querySessionId>
</urn:queryMore>
</soapenv:Body>
</soapenv:Envelope>
```

Example SOAP Response XML

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<queryResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
<result>
<sfobject>
<id>USR-4385</id>
<type>User</type>
<username>test_noexport</username>
</sfobject>
</sfobject>
```



```

<id>USR-4386</id>
<type>User</type>
<username>test_110124102601</username>
</sfoject>
<sfoject>
<id>USR-4387</id>
<type>User</type>
<username>test2_110124102602</username>
</sfoject>
<sfoject>
<id>USR-4388</id>
<type>User</type>
<username>cgrant</username>
</sfoject>
<sfoject>
<id>USR-4389</id>
<type>User</type>
<username>cgrant_2</username>
</sfoject>
<numResults>5</numResults>
<hasMore>>false</hasMore>
<querySessionId>
3209cbb1-2302-41ea-af76-b8158f34bd7d
</querySessionId>
</result>
</queryResponse>
</S:Body>
</S:Envelope>

```

Example SOAP Failed Response

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
<S:Body>
<ns2:Fault xmlns:ns2="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns3="http://
www.w3.org/2003/05/soap-envelope">
<faultcode>ns2:Server</faultcode>
<faultstring>SFAPI Domain Error!</faultstring>
<detail>
<ns2:SFWebServiceFault xmlns="urn:sfoject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
<ns2:errorCode>UNDEFINED_ENTITY_ID</ns2:errorCode>
<ns2:errorMessage>
Entity type 'user23' is undefined!
</ns2:errorMessage>
</ns2:SFWebServiceFault>
</detail>
</ns2:Fault>
</S:Body>
</S:Envelope>

```

Note

The querySessionId returned in the queryResponse is used for the queryMore operation (for pagination), but the server will only keep the last 5 query session ids. Older query session operations will expire. startingRow starts with 1, not 0.

SuccessFactors Query Language

Overview

SFQL (SuccessFactors Query Language) is a SQL-like language used to query data in the SAP SuccessFactors platform. It is used in queryString parameter of the query call. The allowable grammar is documented in BNF Notation (Backus-Naur Form) in the next section.

SFQL Grammar

```
<SFQL_select> ::= SELECT <select_items> FROM <sfapi_object> WHERE <exp>] [order by
<order_by_items>
<order_by_items> ::= <order_by_item> { "," <order_by_item> }*
<order_by_item> ::= <field_name> [ASC | DESC]
<select_items> ::= "*" | <select_item> { "," <select_item> }*
<select_item> ::= <field_name>
<sfapi_object> ::= User | UserRelationship
<exp> ::= <or_exp> | <paren_exp>
<paren_exp> ::= "(" <exp> ")"
<or_exp> ::= <and_exp> { OR <and_exp> }*
<and_exp> ::= <condition> | "(" <or_exp> ")" { AND <condition> | "(" <or_exp>
)" }*
<condition> ::= <sql_condition> | <regular-condition>
<regular_condition> ::= <comparison_item> <comp_op> <comparison_item>
<comp_op> ::= ">" | "<" | "=" | ">=" | "<=" | "<>"
<comparison_item> ::= <simple_exp>
<sql_condition> ::= <in_exp> | <like_exp> | <is_null_exp>
<in_exp> ::= <simple_exp> [ NOT ] IN "(" <simple_exp_list> ")"
<like_exp> ::= <simple_exp> [ NOT ] LIKE <simple_exp>
<is_null_exp> ::= <simple_exp> IS [ NOT ] NULL
<comparison_item> ::= <simple_exp>
<simple_exp> ::= <primary_exp> | "(" <primary_exp> ")"
<simple_exp_list> ::= <simple_exp> { "," <simple_exp> }*
<primary_exp> ::= NULL | <numeric_literal> | <string_literal> | <field_name> |
<date_function_literal>
<date_function_literal> ::= to_date ( <date_string_in_quotes> ,
<date_format_pattern_in_quotes> )
```

Note

The WHERE clause in SFQL is limited to 200 OR/AND subclauses. Due to the Oracle maximum number (1000) of expressions in a list (ORA-01795), SFQL does not accept an IN clause that exceeds 1000 expressions.

4.6 Async Job Operations

Async Job Operations provide an approach to invoke the API asynchronously. Currently the Async Job Operations are limited to running queries on only the Adhoc_XXX entities.

Explain the process flow:

- SubmitQueryJob
- GetJobStatus (explain status values)
- GetJobResults or GetJobResultsEmbedded

API Signature

Signature	Description
TaskStatus submitQueryJob(String queryString, SFParameter[] param)	Submit the asynchronous query job to SAP SuccessFactors with the given query string in SFQL (SuccessFactors Query Language). TaskStatus includes "taskId" which is used to identify a submitted job. For a detailed grammar description, please check the related section in this reference guide.
TaskStatus getJobStatus(String taskId)	Get the execution status of the submitted asynchronous job.
DataHandler getJobResult(GetJobResult job)	Download the result of submitted asynchronous query job.
TaskStatus[] listJobs()	List all Jobs which are in running or waiting to run.
TaskStatus cancelJob(String taskId)	Download the result of submitted asynchronous query job.

API Framework Objects

TaskStatus

TaskStatus has the following properties:

Name	Type	Description
taskId	String	Task Identification. User uses this id to check task status, download task result and cancel a task.
taskName	String	The name of a task. This name can be used to find a task from GUI.
createDate	String	Creation Datetime of a task. The format is ISO-8601. Such as "2012-01-09T03:34:57.964Z"

Name	Type	Description
status	TaskStatusEnum	Enumeration of task status. Status values can be: <ul style="list-style-type: none"> unsubmitted submitted processing finished failed undefined cancelled deleted
taskResult	TaskResultStatus	Object to indicate task result information.

SubmitQueryJob

TaskStatus submitQueryJob(String queryString, SFParameter[] param)

Submit the asynchronous query job to SAP SuccessFactors with the given query string in SFQL (SuccessFactors Query Language). TaskStatus includes "taskId" which is used to identify a submitted job. For a detailed grammar description, please check the related section in this reference guide.

TaskStatus includes the information of submitted task.

Supported Parameters

No parameters for now.

Example Java Code

```

...
    logger.info("[QUERY] Async query started.");
    // Submit Query Job
    TaskStatus taskStatus = sfapi.submitQueryJob(sfql, null);
    logger.info("[QUERY] Async job started. Job Id: " + taskStatus.getTaskId()
        + ". Job Name: " + taskStatus.getTaskName() + ".");
    long startTime = System.currentTimeMillis();
    // Check Query Job status by Polling
    while ((taskStatus.getTaskResult() == null)
        || (taskStatus.getTaskResult().getStatus() !=
            TaskResultStatusEnum.FINISHED)) {

```

```

        if ((taskStatus.getTaskResult() != null)
            && (taskStatus.getTaskResult().getStatus() ==
TaskResultStatusEnum.FAILED)) {
            String msg = "Job failed! Job Id: " + taskStatus.getTaskId()
                + " Job Name: " + taskStatus.getTaskName() + " Error Code: "
                + taskStatus.getTaskResult().getErrorCode() + " Error Message: "
                + taskStatus.getTaskResult().getMessage();
            logger.error(msg);
        }
        // Add error handling code
    }
    long startTime = System.currentTimeMillis();
    if (currentTime - startTime > timeoutMs) {
        logger.error("Connection timeout");
    }
    // Add error handling code
    }
    Thread.sleep(100);
    // Get Job Status
    taskStatus = sfapi.getJobStatus(taskStatus.getTaskId());
}
logger.info("[QUERY] Job finished. Start to get the async query result.");
// Got Job result
DataHandler data = sfapi.getJobResult(
    Constants.FORMAT_ZIPPED-CSV, taskStatus.getTaskId());
FileOutputStream out = new FileOutputStream(new File("queryResult.zip"));
DataHandler d = new DataHandler(ds);
InputStream in = d.getInputStream();
if (out != null && in != null) {
    try {
        int ch = 0;
        while ((ch = in.read()) > 0) {
            out.write(ch);
        }
        out.flush();
    } finally {
        in.close();
        out.close();
    }
}
}
}

```

Example SOAP Request XML

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:submitQueryJob>
      <urn:queryString>
select
emp_personal_info_t_FIRST_NAME,emp_personal_info_t_GENDER,emp_personal_info_t_LAST_N
AME,
    emp_personal_info_t_MIDDLE_NAME
from adhoc_ectint
where emp_personal_info_t_FIRST_NAME is not null
    and emp_personal_info_t_GENDER = 'M'
constrain by filter_dept='all' and filter_div='all' and filter_loc='all'
    and custom01 = 'all' and custom02 = 'all' and custom03 = 'all'
    and custom04 = 'all' and custom07 = 'all' and custom12 = 'all'
    and asOfDate = '2011-10-10'
order by emp_personal_info_t_FIRST_NAME
      </urn:queryString>
    </urn:submitQueryJob>
  </soapenv:Body>
</soapenv:Envelope>

```

Example SOAP Response XML

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <submitQueryJobResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
      <result>
        <taskId>TASK-ADHOC-2364</taskId>
        <taskName>SFAPI_0109172739317_ectint</taskName>
        <createDate>2012-01-09T09:27:41.612Z</createDate>
        <status>submitted</status>
      </result>
    </submitQueryJobResponse>
  </S:Body>
</S:Envelope>
```

GetJobStatus

TaskStatus getJobStatus(String taskId)

Get the execution status of the submitted asynchronous job.

Example Java Code

Please refer to code in above "SubmitQueryJob" section.

Example SOAP Request XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:getJobStatus>
      <urn:taskId>TASK-ADHOC-2364</urn:taskId>
    </urn:getJobStatus>
  </soapenv:Body>
</soapenv:Envelope>
```

Example SOAP Response XML

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
```

```

    <getJobStatusResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
    <result>
    <taskId>TASK-ADHOC-2364</taskId>
    <taskName>SFAPI_0109172739317_ectint</taskName>
    <createDate>2012-01-09T09:27:41.612Z</createDate>
    <status>submitted</status>
    <taskResult>
    <taskResultId>2364</taskResultId>
    <lastFinishTime>2012-01-09T09:28:15.032Z</lastFinishTime>
    <lastStartTime>2012-01-09T09:28:08.266Z</lastStartTime>
    <status>finished</status>
    </taskResult>
    </result>
    </getJobStatusResponse>
  </S:Body>
</S:Envelope>

```

GetJobResult

DataHandler getJobResult(GetJobResult job)

Get data handler of job result, user can use it to get the download stream.

"GetJobResult" has the following attributes:

Name	Type	Description
taskId	String	TaskId which is returned in SubmitQuery-Job
format	String	Currently, it supports csv and zipped-csv format only.

Example Java Code

Please refer to code in above "SubmitQueryJob" section.

Example SOAP Request XML

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:getJobResult>
      <urn:taskId>TASK-ADHOC-1021</urn:taskId>
      <urn:format>csv</urn:format>
    </urn:getJobResult>
  </soapenv:Body>
</soapenv:Envelope>

```

Example SOAP Response

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
X-Powered-By: Servlet 2.4; JBoss-4.0.5.GA (build: CVSTag=Branch_4_0
date=200610162339)/Tomcat-5.5
Content-Type: multipart/related; type="text/xml";
boundary="uuid:996a038b-52d8-4218-9af3-5da7177dbb31"
Transfer-Encoding: chunked
Date: Wed, 25 May 2011 07:58:26 GMT
--uuid:996a038b-52d8-4218-9af3-5da7177dbb31
Content-Type: text/xml
<?xml version='1.0' encoding='UTF-8'?><S:Envelope xmlns:S="http://
schemas.xmlsoap.org/soap/envelope/"><S:Body /></S:Envelope>
--uuid:996a038b-52d8-4218-9af3-5da7177dbb31
Content-Id:<parameters=9ed89f5a-f9a8-4f0d-9bd9-fe5171a4adbb@jaxws.sun.com>
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
"Email", "Username", "ANALYTICS_LIST_VIEW_BUILDER_SUBDOMAIN_ectint_ectint#users_sysinf
o#USERS_SYS_ID"
"kyuan@bestfun.com", "aceadmin", "aceadmin"
"kyuan@bestrun.com", "admin", "admin"
"kyuan@bestrun.com", "admin1", "admin1"
"bpeng@bestrun.com", "admin11", "admin11"
"kyuan@bestrun.com", "admin2", "admin2"
"SystemMessage@bestrun.com", "v4admin", "v4admin"
-uuid:996a038b-52d8-4218-9af3-5da7177dbb31
```

ListJobs

List<TaskStatus> listJobs()

List all running and waiting to run jobs.

Supported Parameters

No parameters for now.

Example Java Code

```
...
    List<TaskStatus> tasks = sfapi.listJobs();
// Iterate tasks
for (TaskStatus task : tasks) {
    System.out.println (task.getId() + " " + task.getName() + " " +
task.getStatus());
}
...

```


Example SOAP Request XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:listJobs/>
  </soapenv:Body>
</soapenv:Envelope>
```

Example SOAP Response XML

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <listJobsResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
      <result>
        <taskId>TASK-ADHOC-2004</taskId>
        <taskName>SFAPI_1101145047.222_ectint</taskName>
        <createDate>2011-11-01T06:50:57.241Z</createDate>
        <status>submitted</status>
        <taskResult>
          <taskResultId>2004</taskResultId>
          <lastFinishTime>2011-11-01T07:14:42.016Z</lastFinishTime>
          <lastStartTime>2011-11-01T07:14:21.232Z</lastStartTime>
          <message>Job Name:[SFAPI_1101145047.222_ectint (2004)] Job Instance
Id:[2004] Process Notes:[Job successfully completed]</message>
          <status>running</status>
        </taskResult>
      </result>
      <result>
        <taskId>TASK-ADHOC-2005</taskId>
        <taskName>SFAPI_1101151557.620_ectint</taskName>
        <createDate>2011-11-01T07:15:57.760Z</createDate>
        <status>submitted</status>
        <taskResult>
          <taskResultId>2005</taskResultId>
          <lastFinishTime>2011-11-01T07:16:28.608Z</lastFinishTime>
          <lastStartTime>2011-11-01T07:16:27.449Z</lastStartTime>
          <message>Job Name:[SFAPI_1101151557.620_ectint (2005)] Job Instance
Id:[2005] Process Notes:[Job successfully completed]</message>
          <status>running</status>
        </taskResult>
      </result>
      <result>
        <taskId>TASK-ADHOC-2006</taskId>
        <taskName>SFAPI_1101152635.919_ectint</taskName>
        <createDate>2011-11-01T07:26:35.959Z</createDate>
        <status>submitted</status>
        <taskResult>
          <taskResultId>2006</taskResultId>
          <lastFinishTime>2011-11-01T07:27:01.349Z</lastFinishTime>
          <lastStartTime>2011-11-01T07:26:59.876Z</lastStartTime>
          <message>Job Name:[SFAPI_1101152635.919_ectint (2006)] Job Instance
Id:[2006] Process Notes:[Job successfully completed]</message>
          <status>running</status>
        </taskResult>
      </result>
    </listJobsResponse>
  </S:Body>
</S:Envelope>
```

```

        </taskResult>
    </result>
    <result>
        <taskId>TASK-ADHOC-2007</taskId>
        <taskName>SFAPI_1101153644.590_ectint</taskName>
        <createDate>2011-11-01T07:36:45.032Z</createDate>
        <status>submitted</status>
        <taskResult>
            <taskResultId>2007</taskResultId>
            <lastFinishTime>2011-11-01T07:37:14.127Z</lastFinishTime>
            <lastStartTime>2011-11-01T07:37:02.420Z</lastStartTime>
            <message>Job Name:[SFAPI_1101153644.590_ectint (2007)] Job Instance
            Id:[2007] Process Notes:[Job successfully completed]</message>
            <status>running</status>
        </taskResult>
    </result>
    <result>
        <taskId>TASK-ADHOC-2008</taskId>
        <taskName>SFAPI_1101153701.852_ectint</taskName>
        <createDate>2011-11-01T07:37:02.007Z</createDate>
        <status>submitted</status>
        <taskResult>
            <taskResultId>2008</taskResultId>
            <lastFinishTime>2011-11-01T07:37:14.127Z</lastFinishTime>
            <lastStartTime>2011-11-01T07:37:02.429Z</lastStartTime>
            <message>Job Name:[SFAPI_1101153701.852_ectint (2008)] Job Instance
            Id:[2008] Process Notes:[Job successfully completed]</message>
            <status>running</status>
        </taskResult>
    </result>
</listJobsResponse>
</S:Body>
</S:Envelope>

```

CancelJob

TaskStatus cancelJob(String taskId)

Cancel a waiting to run job.

Supported Parameters

“taskId” – taskId is the id returned in “submitQueryJob”

Example Java Code

```

...
    TaskStatus task = sfapi.cancelJob("TASK-ADHOC-1200");
    System.out.println (task.getId() + " " + task.getName() + " " + task.getStatus());
...

```

Example SOAP Request XML

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:cancelJob>
      <urn:taskId>TASK-ADHOC-2011</urn:taskId>
    </urn:cancelJob>
  </soapenv:Body>
</soapenv:Envelope>
```

Example SOAP Response XML

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <cancelJobResponse xmlns="urn:sfobject.sfapi.successfactors.com"
xmlns:ns2="urn:fault.sfapi.successfactors.com">
      <result>
        <taskId>TASK-ADHOC-2011</taskId>
        <taskName>SFAPI_1101153813.779_ectint</taskName>
        <createDate>2011-11-01T07:38:15.724Z</createDate>
        <status>canceled</status>
      </result>
    </cancelJobResponse>
  </S:Body>
</S:Envelope>
```

5 User Entity

5.1 Overview

The User entity represents a user account and also contains user demographic and organizational information like a user's department, division, location, job code, manager and other details. The User entity allows data integration between SAP SuccessFactors and your Human Resource Information System (HRIS) or payroll system. This allows single source updates of your employee data. With this data transfer process, the updates you enter into your HRIS will be passed to SAP SuccessFactors, eliminating duplicate entry between both systems. This integration requires the export of the defined data fields (see below). These data can be imported to SAP SuccessFactors either through the API using the "User" entity, or through a flat file, which is uploaded through the Admin Tools user interface. The API can be accessed to upload the data in real time. The file update process can be automated using a Secure File Transport Protocol (SFTP) process executed at your schedule to maintain synchronized data.

5.2 Permissions

Permission access to the User entity is administrative in nature. These permissions grant the API user with access to ALL fields for ALL users in the system. Note that you may limit the scope of users that may be accessed through the file based imports and exports by specifying a target population in either an Administrative Domain (in the traditional permission system) or through a target population in Role Based Permissions. These row restrictions do NOT apply to the API. A user with API login permission AND the administrative permissions of Employee Import and Export will be able to access ALL rows and ALL fields for every user in the system, including the fields from the basic Employee Import/Export and the user personal information fields in the Extended User Information import/export.

The following administrative permissions are required to access the User entity:

API Operation	Permission Required
INSERT	Manage Users --> Employee Import
UPDATE	Manage Users --> Employee Import
UPSERT	Manage Users --> Employee Import
DELETE	You cannot hard delete any row in the User entity. You can only set the status to "inactive" or "inactive_external", which is in effect a soft delete. Note that user data can be purged separately by administrators using the system Data Retention Management functionality.

API Operation	Permission Required
QUERY	Manage Users --> Employee Export

5.3 Field Configuration

The fields in the User entity are configured in the system Data Model, also known as the Succession Data Model. Despite the name, the Succession Data Model is a shared platform data model configuration framework. The Succession Data Model is typically configured by your SAP SuccessFactors implementation representative in consultation with your organization and required business practices. It may also be configured by your company administrator using the SuccessFactory tool.

The User entity will expose every <standard-element> and <userinfo-element> field that appears in the Succession Data Model. The API describeEx() operation will read the field metadata from this data model.

5.4 Differences between API and File Import/Export of Employee Data

There are several differences between importing and exporting of employee data between the API and File based processes.

File based imports and exports of employee data are managed through two separate files:

- (Basic) Employee Import/Export will access the core employee data fields used to create the user account and the core organizational fields required for security and core business processes.
- Extended User Information Import/Export accesses many extended fields used in additional modules like Succession, Compensation and Recruiting.

In the API, all of the fields in the Basic Employee Import/Export AND the Extended User Information field are combined into the User entity.

Field names differ from between the API User entity and the Basic Employee Import. These differences are documented below.

Importing Compensation Field Data

- In the Basic Employee Import file you can add custom fields to the Basic Employee Import file that can be used by the Compensation process, referenced through the Compensation Plan templates. In this file import process, you simply add new columns with new column field headers to the Basic Employee Import file. These new column field headers can have ANY value that is not already reserved for "standard" elements. Your SAP SuccessFactors compensation implementation consultant can guide you on allowable values. The data from

these custom compensation fields will be stored in a staging table indexed with the column name you created in the Basic Employee Import file. These fields can be referenced from Compensation Plan Templates by using the same column id used when importing from the Basic Employee Import file.

- In the API, the compensation fields are loaded in a separate entity called "EmployeeCompensationStagingFields". You cannot import these fields in the User entity in the API. Furthermore, the EmployeeCompensationStagingFields" entity takes a more restrictive approach, by first checking the structure of the Compensation Plan Templates to determine what fields have been configured. For more details on this entity, see the separate documentation on the EmployeeCompensationStagingFields API entity.
- Imports to the User API entity are real time, and are subject to the API batch size rules (maximum batch size of 800). File based imports are unlimited in the file size. File based import of employee data can be scheduled, typically on a nightly basis, so the imports are not real time.
- Date format differs between file imports and API.
- File date formats must be mm/dd/yyyy.
- API date formats are YYYY-MM-DD for fields that are of type "date".
- Some API entities may take a date value tunneled inside a "string" datatype. For example, the EmployeeCompensationStagingFields entity can import custom fields intended for date information, but the data type is string. In this entity, the date formats are "mm/dd/yyyy" and are not validated by the API layer. Please refer to specific entity documentation for exceptions on date formats.

5.5 Business Rules for Data Transfer

Some of the business rules below apply only to the file transfer process, and are indicated as such. We include the file rules here in the API guide because it may be useful during testing to use the manual file upload process to check your API results.

The file based import/export is called "Employee Data Import" and "Employee Data Export" (in non-Employee Central suite configuration) and "Basic User Import/Export" (in Employee Central configuration).

File Based Upload Rules

- The file must be in Comma Separated Values (.csv) format. The file can be a plain text file (.txt) but the values must be separated by commas. Values with commas in them can be enclosed in quotes to load properly.
- Take care when using data fields with leading zeroes. Spreadsheet applications such as Microsoft Excel will strip leading zeroes from data fields and could corrupt the data. If there are leading zeroes in the data file, check before every upload to make sure the leading zeroes are present in the file.
- The file must contain two header rows of predefined column IDs and labels. A template for these column IDs and labels can be exported from the system through the administrative function. In Admin Tools, see Employee Export & Export Template
- STATUS must be the first field and USERID must be the second field and the upper case header must be repeated for the second row. Remaining fields need not be in the order presented here.

API and File Based Rules

The following rules apply to both API and file import of employee data. In the API import, this is done through the User entity. Note that the field names in the API vary slightly from the file field names. For example, the department API field name is called DEPARTMENT in the file import. The file import field names can be seen by exporting the file template in admin tools.

- Throughout this document we reference the API field names.
- The system has a base set of fields that are required, including the following: externalId, username and status. In addition to these fields, your system can be configured to require additional fields. In a table following this section, we make a recommendation on configuring extra fields as required since these fields can be used in the system for reporting and managing security.
- If a field will not be populated, do not include that field in the API feed or the import file. If a field is included in either the API or the import file, and the field is blank (or if it is listed as null in the API) it will wipe out what is already in the database. Simply omit the field if you do not want to update it.
- All String field values are case sensitive. If the username is loaded with mixed case, the logon process will require mixed case.
- The username field must be a unique name of each user and is used as the logon ID value. This value is visible in a variety of places to all end users. Accordingly, username should not contain data that is considered confidential, such as social security number. Unlike externalUserId, this value can be updated if needed, (e.g. name change) but must be unique. Most customers choose to use the same username that they use for their local network or email application.
- The externalId field must be unique for each user, and is used as a system field to link records. This value is visible in a variety of places to all end users. Accordingly, externalId should not contain data that is considered confidential, such as social security number. This value must be passed with each employee data record during each upload. Take care in selecting this value as it this is a permanent decision. This value cannot be changed once set.
- Since the jobCode field is a String and is case sensitive, it must match exactly the case sensitivity of job codes designated for job roles. In Admin Tools, see ► [Managing Competencies and Skills](#) ► [Families and Roles](#) ►. The jobCode is used to determine competency mapping which allows competencies to be populated on forms and worksheets based on the value in the JOBCODE field. If the company is using job profiles, the value entered in this column will determine what set of competencies will be displayed on the appraisal form for that employee. This value may be the actual job code from the HRIS or a derived code. Many customers will choose to combine job codes into broad employee groups to simplify administration of competency sets that may be common across many roles in the same group. This decision should be made by the implementation team based on the process.
- The ability for the employee to edit/not edit the data loaded into the employee record (on the Personal Information screen) is an option defined during initial set-up. We strongly recommend that these edit rights be turned off for any values you are importing from your HRIS system. This is necessary to maintain sync of data with the HRIS. If you allow a field to be editable in the user interface and you also import this field from your HRIS, then the user edits via the user interface will be overwritten by the next data load from the HRIS.
- The individual fields in the employee record can also be hidden during the initial setup so they will not display in the Personal Information screen. This allows for the storage of data associated with each individual, passed from the HRIS to be used by one or more process, but does not display this data on the Personal Information screen (accessed by the employee).
- If the department, division and/or location fields are not used, do not include them in the import file. The system will automatically assign "N/A" values in these fields. Security and report filtering settings are tied to

these values and entering a common value (e.g. NA) will allow for the broadest reporting access to the data. Leaving it blank will limit reporting and security capabilities.

- Data placed in the Department, Division, and Location fields are String values. They may be a code or a text name. The value entered will appear in drop-down menus.
- Field label values are limited to 100 characters. Field labels in the API are viewed through the describeEx() metadata, and are presented in the locale of the API user. In the file imports/exports, the field label values are presented as the second line header row. You cannot modify the field label values through the API or through the file feeds. The field label values are localizable, and are configured through a separate system setup process.
- Most String field data values are limited to 255 characters.
- The hireDate field is used in SAP SuccessFactors both for simple display purposes, and also can be the date used to initiate an automatic form creation for this user. The value in the hireDate field does not have to be the actual hire date and may be a derived date from the HRIS system (e.g. hire date + 90 days, last review date - 60 days, etc.) to drive when a new document will be created for this employee. The system can be set to automatically generate a new form for this employee based upon the month/day combination of this field value (i.e. if Jim's hire date is 03/02/2002, the system can be set to automatically create a new appraisal form for Jim each March 2nd).
- Additional custom fields can be defined to import additional information (e.g. FLSA code, etc.) using what we call "Userinfo" custom fields. However, this data does not show up anywhere in the system, other than the Employee Profile or custom fields in a form. They can also be used in Organization Chart. See the general section on custom field configuration above.
- Employee Relationship Fields: There are five user to user relationship types that can be used for managing security, workflows and reporting. These five relationships and their field names are described below.
- The Manager relationship is used to create the employee hierarchy. Loops are prevented (meaning a manager cannot report to themselves). Each employee must have a valid manager. Upon data load, SAP SuccessFactors will validate that each employee has a valid manager listed in their data record. This validation check is executed against the employees held within the SAP SuccessFactors database. Therefore, the manager for each user record must refer to another existing user in the database. This validation is done within a single API batch operation. This means you can insert new records for both an employee and their manager in a single API batch call. The manager can be set through two API fields: The managerExternalId field maps to the externalId of the User record for the employee's manager, and the managerId field which maps to the id field of the manager, which is an internal Id value. We suggest using the managerExternalId for simplicity. Do not supply both values.
- The HR Manager is intended for security settings to allow the HR manager to access the user. The HR Manager is not enforced as a hierarchy. The HR manager can be set through two API fields: The hrExternalId field maps to the externalId of the HR manager, and the hrId field which maps to the id field of the HR manager, which is an internal Id value. We suggest using the hrExternalId for simplicity. Do not supply both values.
- The Second Manager relationship is another employee hierarchy, and loops are prevented (a manager cannot report to himself). This is typically used in compensation planning processes to freeze the hierarchy at a point in time from daily changes, to allow a focal review process against an unchanging employee hierarchy. The API provides only one field for this relationship, the secondManagerExternalId which maps to the externalId of the User record for the second manager.
- The Matrix Manager and Custom Manager relationships are many-to-many, meaning a user can have multiple matrix or custom managers. The fields are matrixManagerExternalIds and customManagerExternalIds, which map to the User.externalId values of the intended matrix and custom manager User records. The syntax for these fields is to separate the manager IDs with pipe (|) characters. Example: "gsmith624|sholmes423|smaddox666".

- Best practice security integration recommendation: The Matrix Manager and Custom Manager relationships can also be loaded separately from the User entity in the MatrixManager and CustomManager API entities. This is useful to increase performance of the loads by providing delta updates instead of full load and replace of all relationships.
- Best practice security integration recommendation: Each of the five relationships above are highly useful in translating between your HRIS organizational security structures into the SAP SuccessFactors system, which does not have organizational structures. In the current SAP SuccessFactors security capabilities, your data feeds from the HRIS will typically explode out your organizational security access into a person to person mapping that will be loaded into these structures. For example, if your HRIS has a department hierarchy where Dept1 contains Dept2 and Dept3, and you want to grant an HR person with access to the Dept1 hierarchy, you can explode out the users that are assigned to the Dept1 HRIS hierarchy, and assign each of these users to have an HR Manager, Matrix Manager, or Custom Manager as the HR person who needs access to these users.
- Best practice security integration recommendation: An alternative to using the above relationships to translate from your HRIS organizational security structures is to use one of the CUSTOM01-15 fields to contain structural identifiers that can group people into substructures of your HRIS organizational structures. This is a translation from your HRIS hierarchical structures into flat identifiers, or tags. For example, if your organizational structure has a hierarchy where Dept1 contains Dept2 and Dept3, and you grant users access to the Dept1 hierarchy in your HRIS, you can tag all users in this hierarchy with a value of "Dept1-Hierarchy" in the CUSTOM01 field in SAP SuccessFactors, and grant security access rules through this tag in SAP SuccessFactors. Then if "Dept4" is added to the HRIS Dept1 hierarchy, your data feed will update the appropriate users with the "Dept1-Hierarchy" tag who match your new Dept4 organization. You can of course, also assign access directly to the departments field in SAP SuccessFactors, but if new departments are added in the future, you will have to update security settings in SAP SuccessFactors to grant access to them as well. This process will depend on how your organization is managed.
- For both the Manager and Custom Manager hierarcies, the individual at the top of the employee hierarchy (e.g. CEO) must have the value of "NO_MANAGER" (in all caps) listed in the managerExternalId field, so that the system knows how to treat this individual in the routing chain. This value must be exactly NO_MANAGER. Other values, such as NO_MGR will fail.
- Since the data load validation process validates the managerExternalId and hrExternalId against the existing database, it is strongly recommended that the new employees be added to the beginning of the data load file. This will allow the new employee to be loaded first in case the new employee is a manager or HR representative referenced later in the data file and will improve processing speed.
- The country field can be used to determine which of several possible Privacy Consent Statements will be presented to a new user. If the Privacy Consent feature is enabled, when users log in to the system the very next time, they will go directly to the data consent page - which can differ depending on their COUNTRY. Users can then choose to accept or decline the terms presented. Use the country names specified in the Country List at the end of this document.
- The defaultLocale field is not typically required. If present, the defaultLocale will determine the languages experienced by the users (one language per user). Users can specify their locale preferences in user options. See the list of supported locales at the end of this document.
- The proxyExternalIds field defines which persons can act as proxy on behalf of the user. If the field is included, and the values are null or blank, no proxy holder will be assigned and existing proxy holders will be removed. If the field is omitted, the field values for this user will be unaffected (this is an API standard). The field can include multiple proxy holders. The syntax is to separate the proxy holder IDs with pipe (|) characters. Example: "admin|sholmes423|smaddox666", where the proxy holder references are the User.externalId values. Granting proxy through this field will give the proxy holder full proxy rights, you cannot control "Proxy by module" granular details. (Proxy by Module details allow you to specify which module a proxy holder may access when proxying in for a user). This capability may be added in the future in a separate API entity. In the

meantime, SAP SuccessFactors does provide a file import format for this Proxy By Modeul, and the file import can be automated.

- The timeZone field is recommended in the system for internal time/date stamps in the time of the user. See the “Supported Time Zones” section at the end of this document. If a timeZone value is not loaded the field will be automatically populated with a default of Eastern Standard Time. Records may be set to one value (e.g. time zone of corporation) rather than different zones for each employee. To have the system automatically adjust for daylight savings time use the time zone ID from the Supported Time Zones section at the end of this document rather than the short three-letter abbreviation.

5.6 Common Fields in HRIS Integration

Below is a table of common User entity fields used when integrating from your HRIS system. The User entity can contain many more fields than documented here. The SAP SuccessFactors system is flexible and allows you to enable or disable a large set of standard fields, and also create many custom fields for the User entity. For the full list of fields configured in your system, use the API describeEx() operation on the User entity. You may also consult the API Data Dictionary tool in Admin tools to view the structure of the User entity in your system. The choice of what data to import in the User entity will be a configuration and integration decision based on your specific processes. The list below represents the common base fields most systems use for provisioning users and synchronizing the core employee organizational data from your HRIS system.

API Field Name	File Column Name	Required	Data Type	Example	Functional Area Using this Data
status	STATUS	TRUE	STRING	ACTIVE INACTIVE ACTIVE_EXTERNAL INVACTIVE_EXTERNAL	Indicates the status of the employee. User records cannot be deleted, you can only set them to inactive. Inactive records are not part of business processes, though they can be queried or exported by the API, analytics and administrative functions. External status values are visible only to the 360 module, and Administrative functions

API Field Name	File Column Name	Required	Data Type	Example	Functional Area Using this Data
externalId	USERID	TRUE	STRING	BOBJONES	SYSTEM ID – unique ID. See notes on this field above.
username	USERNAME	TRUE	STRING	BOBJONES	Log-on name/ID. Case sensitive. Users must provide the same case sensitivity during logon. See notes on this field above.
firstName	FIRSTNAME	TRUE	STRING	Bob	User Information
lastName	LASTNAME	TRUE	STRING	Jones	User Information
middleName	MI	FALSE	STRING	P	User Information
gender	GENDER	TRUE	STRING	M or F	Writing Assistant
email	EMAIL	TRUE	STRING	bj@123.com [HYPERLINK REMOVED]	E-mail Notifications
managerExternalId	MANAGER	TRUE	STRING	SSMITH	Workflow routing, Reporting, Security, User Information.
hrExternalId	HR	TRUE	STRING	DDAVIDSON	HR Approval, Routing, Security, Reporting. See notes on this field above.
department	DEPARTMENT	RECOMM	STRING	MARKETING	User Information, Reporting. Default value of N/A will be assigned if no value is provided.
jobCode	JOBCODE	RECOMM	STRING	321	Competency Mapping; User Info

API Field Name	File Column Name	Required	Data Type	Example	Functional Area Using this Data
division	DIVISION	RECO M	STRING	Mkt	Reporting, Security. Default value of N/A will be as- signed if no value is provided.
location	LOCATION	RECO M	STRING	SM	Reporting, Security. Default value of N/A will be as- signed if no value is provided.
timeZone	TIMEZONE	TRUE	STRING	PST	Time Stamp
hireDate	HIREDATE	FALSE	DATE	API: 2001-01-01 File: 1/1/2001	Date of Company Hire (Also can be used to trigger auto creation of forms)
employeeId	EMPID	FALSE	STRING	1234	User Information
title	TITLE	FALSE	STRING	ANALYST	User Information
businessPhone	BIZ_PHONE	FALSE	STRING	6504755123	User Information
businessFax	FAX	FALSE	STRING	6504755124	User Information
addressLine1	ADDR1	FALSE	STRING	2 MAPLE ST	User Information
addressLine2	ADDR2	FALSE	STRING	SUITE 2	User Information
city	CITY	FALSE	STRING	SAN MATEO	User Information
state	STATE	FALSE	STRING	CA	User Information
zipCode	ZIP	FALSE	STRING	94403	User Information
country	COUNTRY	FALSE	STRING	USA	Data Privacy Con- sent Statements
reviewFrequency	REVIEW_FREQ	FALSE	STRING	1	Not needed

API Field Name	File Column Name	Required	Data Type	Example	Functional Area Using this Data
lastReviewDate	LAST_REVIEW_DATE	FALSE	DATE	API: 2012-04-19 File: 4/19/2012	User Information
custom01	CUSTOM01	FALSE	STRING	Any string value	Used to set permissions, filters in reporting, and displayed in forms, Employee Profile, Organization Chart. See notes on this field above.
custom02	CUSTOM02	FALSE	STRING	Any string value	Same as CUSTOM01
custom03	CUSTOM03	FALSE	STRING	Any string value	Same as CUSTOM01
custom04	CUSTOM04	FALSE	STRING	Any string value	Same as CUSTOM01
custom05	CUSTOM05	FALSE	STRING	Any string value	Same as CUSTOM01
custom06	CUSTOM06	FALSE	STRING	Any string value	Same as CUSTOM01
custom07	CUSTOM07	FALSE	STRING	Any string value	Same as CUSTOM01
custom08	CUSTOM08	FALSE	STRING	Any string value	Same as CUSTOM01
custom09	CUSTOM09	FALSE	STRING	Any string value	Same as CUSTOM01
custom10	CUSTOM10	FALSE	STRING	Any string value	Same as CUSTOM01
custom11	CUSTOM11	FALSE	STRING	Any string value	Same as CUSTOM01
custom12	CUSTOM12	FALSE	STRING	Any string value	Same as CUSTOM01
custom13	CUSTOM13	FALSE	STRING	Any string value	Same as CUSTOM01

API Field Name	File Column Name	Required	Data Type	Example	Functional Area Using this Data
custom14	CUSTOM14	FALSE	STRING	Any string value	Same as CUSTOM01
custom15	CUSTOM15	FALSE	STRING	Any string value	Same as CUSTOM01
matrixManagerExternalIds	MATRIX_MANAGER	FALSE	STRING	SSMITH	Dotted line relationship. See notes on this field above.
defaultLocale	DEFAULT_LOCALE	FALSE	STRING	fr_FR	Language seen by user. See the Related Information section for a list of languages and locales supported.
customManagerExternalIds	CUSTOM_MANAGER	FALSE	STRING	SSMITH	Dotted line relationship/Succession Planner. See notes on this field above.
secondManagerExternalId	SECOND_MANAGER	FALSE	STRING	SSMITH	Dotted line relationship/Compensation Manager. See notes on this field above.
proxyExternalIds	PROXY	FALSE	STRING	SSMITH	Assign proxy(ies) for user. See notes on this field above.

Fields can be made required or not required to reflect business practices. RECOM means we recommend that the field be made required. Choice of required or not required is specified when the field is configured in the data model. See the field configuration section for details on how to configure fields.

Special Text Field Character Limitations

Character(s)	Description	User Id	Username	Password
a thru z	Lowercase letters	Y	Y	Y
A thru Z	Uppercase letters	Y	Y	Y

Character(s)	Description	User Id	Username	Password
0 thru 9	Numerals	Y	Y	Y
_	Underscore	Y	Y	Y
-	Hyphen	Y	Y	Y
@	At sign	Y	Y	Y
.	Period	Y	Y	Y
~	Tilde	N	Y	Y
!	Exclamation mark	N	Y	Y
#	Pound sign	N	Y	Y
\$	Dollar sign	N	Y	Y
%	Percent sign	N	Y	Y
&	Ampersand	N	Y	Y
*	Asterisk	N	Y	Y
+	Plus sign	N	Y	Y
'	Single quote	N	Y	N
‘	Accent grave	N	Y	N
?	Question mark	N	Y	Y
=	Equal sign	N	N	Y
^	Caret	N	N	Y
(Left Parenthesis	N	N	Y
)	Right Parenthesis	N	N	Y
,	Comma	N	N	Y
/	Slash	N	N	N
;	Semicolon	N	N	Y
[Left bracket	N	N	Y
]	Right bracket	N	N	Y
<	Less than	N	N	Y

Character(s)	Description	User Id	Username	Password
>	Greater than	N	N	Y
:	Colon	N	N	Y
“	Double quote	N	N	N
{	Left braces	N	N	Y
}	Right braces	N	N	Y
Back slash	Back slash	N	N	Y
Pipe sign	Pipe sign	N	N	Y
	Space	N	N	N

Related Information

[SAP SuccessFactors Languages and Locales](#)

5.7 Optional Business Processing Parameters

There are several optional processing parameters that can control business logic when editing rows in the User entity. For example, the User entity allows you to change the manager for a user. When changing a user's manager, you can tell the system how to handle any documents that may be in the user's inbox. Documents include Performance Review Forms and 360's. In the file-based import, the user interface presents the options on the Employee Import page in the Administrator Tools.

You can add the following parameters when editing a user through insert, update, or upsert operations with the User entity.

Parameter	Value	Description
defaultPasswordField	USERNAME externalId EMAIL FIRSTNAME LASTNAME RANDOM_PWD	<p>Use this parameter to set the password to a corresponding field value when creating new users. Note that you can also set a password value directly by specifying the password field. The value specified in the password field takes precedence over the parameter value. When no parameter value or password field value is specified, the system uses user name as the default password.</p> <div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <p>⚠ Caution</p> <p>Parameter values USERNAME, externalId, EMAIL, FIRSTNAME, and LASTNAME are unsafe password values. We recommend that you use system-generated random passwords (RANDOM_PWD) when creating users.</p> <p>You can enable the Use System Generated Password by Default option in Admin Center > Platform Feature Settings to force all new and existing API requests to use system-generated random passwords except for the ones with values specified in the password field. Note that you need to review your existing API calls to make sure they continue to work after enabling the option.</p> </div>
managerOnlyCompany	true / false	<p>Enforce Manager-only Implementation. This feature isn't supported when Role-Based Permissions are used.</p>

Parameter	Value	Description
managerTransfer	true / false	Automatic insertion of new manager as next document recipient if not already. This means that the new manager will see the document when the current step owner moves the document to the next step in the workflow.
matrixManagerTransfer	true / false	Matrix Manager – Automatic insertion of new manager as next document recipient if not already
processCompletedDocsToManager	true / false	Manager – Automatic Process Owner Change To New Manager For Completed Documents When Old Manager is Process Owner (Only for 360)
processCompletedDocsToMatrixManager	true / false	Matrix Manager – Automatic Process Owner Change To New Manager For Completed Documents When Old Manager is Process Owner (Only for 360)
processInactiveEmployees	true / false	Process inactive Employees
processInProgressDocsToManager	true / false	Manager – Automatic Process Owner Change To New Manager For In-Progress Documents When Old Manager is Process Owner (Only for 360)
processInProgressDocsToMatrixManager	true / false	Matrix Manager – Automatic Process Owner Change To New Manager For In-Progress Documents When Old Manager is Process Owner (Only for 360)
remove360EvaluationDocsForInactiveUsers	true / false	Remove Inactive Employees' 360 Evaluation Documents

Parameter	Value	Description
removeCompletedDocsForInactiveUsers	true / false	Remove Inactive Employees' Completed Documents
removeInProgressDocsForInactiveUsers	true / false	Remove Inactive Employees' In-Progress Documents
routeCompletedDoc	true / false	Automatic Completed Document Copy to New Manager
routeCompletedDocToMatrixManager	true / false	Matrix Manager – Automatic Completed Document Copy to New Manager
routeEnRouteDoc	true / false	Automatic En Route Document Transfer To New Manager
routeEnRouteDocToMatrixManager	true / false	Matrix Manager – Automatic En Route Document Transfer To New Manager
routeInboxDoc	true / false	Automatic Inbox Document Transfer To New Manager
routeInboxDocToMatrixManager	true / false	Matrix Manager – Automatic Inbox Document Transfer To New Manager
sendWelcomeMessage	true / false	Send welcome message to new users

ⓘ Note

Set the parameter value to **true** if you choose to use system-generated password when creating new users so that users can receive e-mail notifications and set their passwords. This parameter is ignored and e-mail notifications are sent to new users if the *Send Welcome Message* option is enabled in [Admin Center](#) > [Platform Feature Settings](#).

Parameter	Value	Description
validateMgrHr	true / false	Check if the Manager and HR fields are valid (meaning that they're active users in the system)

5.8 Example Upsert With SFParameter

The following example shows how to specify the managerTransfer option when UPSERTING a user entity. Note the processingParam XML element below highlighted in yellow.

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:upsert>
      <urn:type>User</urn:type>
      <!--1 or more repetitions:-->
      <urn:sfobject>
        <urn:type>User</urn:type>
        <!--You may enter ANY elements at this point-->
        <externalId>scolbert</externalId>
        <status>ACTIVE</status>
        <username>scolbert</username>
        <firstName>Steven</firstName>
        <lastName>Colbert</lastName>
        <gender>M</gender>
        <division>ACE Software</division>

        <department>Sales</department>
        <location>West</location>
        <email>scolbert@123.com</email>
        <hrExternalId>athompson</hrId>
        <managerExternalId>cgrant</managerId>
        <timeZone>US/Pacific</timeZone>
        <password>pwd</password>
      </urn:sfobject>
      <!--1 or more repetitions:-->
      <urn:processingParam>
        <urn:name>managerTransfer</urn:name>
        <urn:value>true</urn:value>
      </urn:processingParam>
    </urn:upsert>
  </soapenv:Body>
</soapenv:Envelope>
```

5.9 Time Zone Values

Following Time Zone values are supported:

US/Alaska	Africa/Brazzaville	Africa/Libreville
US/Aleutian	Africa/Bujumbura	Africa/Lome
US/Arizona	Africa/Cairo	Africa/Luanda
US/Central	Africa/Casablanca	Africa/Lubumbashi
US/Eastern	Africa/Ceuta	Africa/Lusaka
US/Hawaii	Africa/Conakry	Africa/Malabo
US/Michigan	Africa/Dakar	Africa/Maputo
US/Mountain	Africa/Dar_es_Salaam	Africa/Maseru
US/Pacific	Africa/Djibouti	Africa/Mbabane
US/Samoa	Africa/Douala	Africa/Mogadishu
Africa/Abidjan	Africa/EL_Aaiun	Africa/Monrovia
Africa/Accra	Africa/Freetown	Africa/Nairobi
Africa/Addis_Ababa	Africa/Gaborone	Africa/Ndjamena
Africa/Algiers	Africa/Harare	Africa/Niamey
Africa/Asmara	Africa/Johannesburg	Africa/Nouakchott
Africa/Bamako	Africa/Kampala	Africa/Ouagadougou
Africa/Bangui	Africa/Khartoum	Africa/Porto-Novo
Africa/Banjul	Africa/Kigali	Africa/Sao_Tome
Africa/Bissau	Africa/Kinshasa	Africa/Tripoli
Africa/Blantyre	Africa/Lagos	Africa/Tunis
Africa/Windhoek	America/Danmarkshavn	America/Managua
America/Adak	America/Dawson	America/Manaus
America/Anchorage	America/Dawson_Creek	America/Marigot

America/Anguilla	America/Denver	America/Martinique
America/Antigua	America/Detroit	America/Mazatlan
America/Araguaina	America/Dominica	America/Menominee
America/Argentina/Buenos_Aires	America/Edmonton	America/Merida
America/Argentina/Catamarca	America/Eirunepe	America/Mexico_City
America/Argentina/Cordoba	America/El_Salvador	America/Miquelon
America/Argentina/Jujuy	America/Fortaleza	America/Moncton
America/Argentina/La_Rioja	America/Glace_Bay	America/Monterrey
America/Argentina/Mendoza	America/Godthab	America/Montevideo
America/Argentina/Rio_Gallegos	America/Goose_Bay	America/Montreal
America/Argentina/San_Juan	America/Grand_Turk	America/Montserrat
America/Argentina/San_Luis	America/Grenada	America/Nassau
America/Argentina/Tucuman	America/Guadeloupe	America/New_York
America/Argentina/Ushuaia	America/Guatemala	America/Nipigon
America/Aruba	America/Guayaquil	America/Nome
America/Asuncion	America/Guyana	America/Noronha
America/Atikokan	America/Halifax	America/North_Dakota/Center
America/Bahia	America/Havana	America/North_Dakota/New_Salem
America/Barbados	America/Hermosillo	America/Panama
America/Belem	America/Indiana/Indianapolis	America/Pangnirtung
America/Belize	America/Indiana/Knox	America/Paramaribo
America/Blanc-Sablon	America/Indiana/Marengo	America/Phoenix
America/Boa_Vista	America/Indiana/Petersburg	America/Port-au-Prince
America/Bogota	America/Indiana/Tell_City	America/Port_of_Spain
America/Boise	America/Indiana/Vevay	America/Porto_Velho
America/Cambridge_Bay	America/Indiana/Vincennes	America/Puerto_Rico
America/Campo_Grande	America/Indiana/Winamac	America/Rainy_River

America/Cancun	America/Inuvik	America/Rankin_Inlet
America/Caracas	America/Iqaluit	America/Recife
America/Cayenne	America/Jamaica	America/Regina
America/Cayman	America/Juneau	America/Resolute
America/Chicago	America/Kentucky/Louisville	America/Rio_Branco
America/Chihuahua	America/Kentucky/Monticello	America/Santiago
America/Costa_Rica	America/La_Paz	America/Santo_Domingo
America/Cuiaba	America/Lima	America/Sao_Paulo
America/Curacao	America/Los_Angeles	America/Scoresbysund
America/St_Barthelemy	America/Maceio	America/Shiprock
America/St_Johns	Asia/Brunei	Asia/Rangoon
America/St_Kitts	Asia/Choibalsan	Asia/Riyadh
America/St_Lucia	Asia/Chongqing	Asia/Sakhalin
America/St_Thomas	Asia/Colombo	Asia/Samarkand
America/St_Vincent	Asia/Damascus	Asia/Seoul
America/Swift_Current	Asia/Dhaka	Asia/Shanghai
America/Tegucigalpa	Asia/Dili	Asia/Singapore
America/Thule	Asia/Dubai	Asia/Taipei
America/Thunder_Bay	Asia/Dushanbe	Asia/Tashkent
America/Tijuana	Asia/Gaza	Asia/Tbilisi
America/Toronto	Asia/Harbin	Asia/Tehran
America/Tortola	Asia/Ho_Chi_Min	Asia/Thimphu
America/Vancouver	Asia/Hong_Kong	Asia/Tokyo
America/Whitehorse	Asia/Hovd	Asia/Ulaanbaatar
America/Winnipeg	Asia/Irkutsk	Asia/Urumqi
America/Yakutat	Asia/Jakarta	Asia/Vientiane
America/Yellowknife	Asia/Jayapura	Asia/Vladivostok

Antarctica/Casey	Asia/Jerusalem	Asia/Yakutsk
Antarctica/Davis	Asia/Kabul	Asia/Yekaterinburg
Antarctica/DumontDURville	Asia/Kamchatka	Asia/Yerevan
Antarctica/Mawson	Asia/Karachi	Atlantic/Azores
Antarctica/McMurdo	Asia/Kashgar	Atlantic/Bermuda
Antarctica/Palmer	Asia/Katmandu	Atlantic/Canary
Antarctica/Rothera	Asia/Kolkata	Atlantic/Cape_Verde
Antarctica/South_Pole	Asia/Krasnoyarsk	Atlantic/Faroe
Antarctica/Syowa	Asia/Kuala_Lumpur	Atlantic/Madeira
Antarctica/Vostok	Asia/Kuching	Atlantic/Reykjavik
Arctic/Longyearbyen	Asia/Kuwait	Atlantic/South_Georgia
Asia/Aden	Asia/Macau	Atlantic/St_Helena
Asia/Almaty	Asia/Magadan	Atlantic/Stanley
Asia/Amman	Asia/Makassar	Australia/Adelaide
Asia/Anadyr	Asia/Manila	Australia/Brisbane
Asia/Aqtau	Asia/Muscat	Australia/Broken_Hill
Asia/Aqtobe	Asia/Nicosia	Australia/Currie
Asia/Ashgabat	Asia/Novosibirsk	Australia/Darwin
Asia/Baghdad	Asia/Omsk	Australia/Eucla
Asia/Bahrain	Asia/Oral	Australia/Hobart
Asia/Baku	Asia/Phnom_Penh	Australia/Lindeman
Asia/Bangkok	Asia/Pontianak	Australia/Lord_Howe
Asia/Beirut	Asia/Pyongyang	Australia/Melbourne
Asia/Bishkek	Asia/Qatar	Australia/Perth
Europe/Amsterdam	Asia/Qyzylorda	Australia/Sydney
Europe/Andorra	Europe/Rome	Pacific/Efate
Europe/Athens	Europe/Samara	Pacific/Enderbury

Europe/Belgrade	Europe/San_Marino	Pacific/Fakaofu
Europe/Berlin	Europe/Sarajevo	Pacific/Fiji
Europe/Bratislava	Europe/Simferopol	Pacific/Funafuti
Europe/Brussels	Europe/Skopje	Pacific/Galapagos
Europe/Bucharest	Europe/Sofia	Pacific/Gambier
Europe/Budapest	Europe/Stockholm	Pacific/Guadalcanal
Europe/Chisinau	Europe/Tallinn	Pacific/Guam
Europe/Copenhagen	Europe/Tirane	Pacific/Honolulu
Europe/Dublin	Europe/Uzhgorod	Pacific/Johnston
Europe/Gibraltar	Europe/Vaduz	Pacific/Kiritimati
Europe/Guernsey	Europe/Vatican	Pacific/Kosrae
Europe/Helsinki	Europe/Vienna	Pacific/Kwajalein
Europe/Isle_of_Man	Europe/Vilnius	Pacific/Majuro
Europe/Istanbul	Europe/Volgograd	Pacific/Marquesas
Europe/Jersey	Europe/Warsaw	Pacific/Midway
Europe/Kaliningrad	Europe/Zagreb	Pacific/Nauru
Europe/Kiev	Europe/Zaporozhye	Pacific/Niue
Europe/Lisbon	Europe/Zurich	Pacific/Norfolk
Europe/Ljubljana	Indian/Antananarivo	Pacific/Noumea
Europe/London	Indian/Chagos	Pacific/Pago_Pago
Europe/Luxembourg	Indian/Christmas	Pacific/Palau
Europe/Madrid	Indian/Cocos	Pacific/Pitcairn
Europe/Malta	Indian/Comoro	Pacific/Ponape
Europe/Mariehamn	Indian/Kerguelen	Pacific/Port_Moresby
Europe/Minsk	Indian/Mahe	Pacific/Rarotonga
Europe/Monaco	Indian/Maldives	Pacific/Saipan
Europe/Moscow	Indian/Mauritius	Pacific/Tahiti

Europe/Oslo	Indian/Mayotte	Pacific/Tarawa
Europe/Paris	Indian/Reunion	Pacific/Tongatapu
Europe/Podgorica	Pacific/Apia	Pacific/Truk
Europe/Prague	Pacific/Auckland	Pacific/Wake
Europe/Riga	Pacific/Chatham	Pacific/Wallis
	Pacific/Easter	

5.10 Deprecated Time Zone Values

The following time zones are supported for backwards compatibility. They are deprecated since they can lead to confusion. For example, does CST represent Central Standard Time or China Standard Time?

ACT	BET	CST	EST	JST	NST	PST	VST
AET	BST	CTT	GMT	MET	PLT	ROK	WET
AGT	CAT	EAT	HST	MIT	PNT	SST	
ART	CET	ECT	IET	MST	PRC	UCT	
AST	CNT	EET	IST	NET	PRT	UTC	

5.11 Supported Countries and Regions

The following values are supported for the country field, which can be used for data privacy and purge rules.

Afghanistan	Cameroon	Ethiopia
Albania	Canada	Fiji
Algeria	CapeVerde	Finland
Andorra	CentralAfricanRepublic	France
Angola	Chad	Gabon

AntiguaandBarbuda	Chile	Gambia,The
Argentina	China	Georgia
Armenia	Colombia	Germany
Australia	Comoros	Ghana
Austria	Congo(Brazzaville)	Greece
Azerbaijan	Congo,DemocraticRepublicofthe	Grenada
Bahamas	CostaRica	Guatemala
Bahrain	Côted'Ivoire	Guinea
Bangladesh	Croatia	Guinea-Bissau
Barbados	Cuba	Guyana
Belarus	Cyprus	Haiti
Belgium	CzechRepublic	Honduras
Belize	Denmark	Hong Kong
Benin	Djibouti	Hungary
Bhutan	Dominica	Iceland
Bolivia	DominicanRepublic	India
BosniaandHerzegovina	EastTimor(TimorTimur)	Indonesia
Botswana	Ecuador	Iran
Brazil	Egypt	Iraq
Brunei	ElSalvador	Ireland
Bulgaria	EquatorialGuinea	Israel
BurkinaFaso	Eritrea	IvoryCoast
Burundi	Estonia	Italy
Cambodia	Namibia	Jamaica
Japan	Nauru	SolomonIslands
Jordan	Nepal	Somalia
Kazakhstan	Netherlands	SouthAfrica

Kenya	NewZealand	Spain
Kiribati	Nicaragua	SriLanka
Korea,North	Niger	Sudan
Korea,South	Nigeria	Suriname
Kuwait	Norway	Swaziland
Kyrgyzstan	Oman	Sweden
Laos	Pakistan	Switzerland
Latvia	Palau	Syria
Lebanon	Panama	Taiwan
Lesotho	PapuaNewGuinea	Tajikistan
Liberia	Paraguay	Tanzania
Libya	Peru	Thailand
Liechtenstein	Philippines	Togo
Lithuania	Poland	Tonga
Luxembourg	Portugal	TrinidadandTobago
Macedonia,FormerYugoslavRepublicof	PuertoRico	Tunisia
Madagascar	Qatar	Turkey
Malawi	Romania	Turkmenistan
Malaysia	Russia	Tuvalu
Maldives	Rwanda	Uganda
Mali	SaintKittsandNevis	Ukraine
Malta	SaintLucia	UnitedArabEmirates
MarshallIslands	SaintVincentandTheGrenadines	UnitedKingdom
Mauritania	Samoa	UnitedStates
Mauritius	SanMarino	Uruguay
Mexico	SaoTomeandPrincipe	Uzbekistan
Micronesia,FederatedStatesof	SaudiArabia	Vanuatu

Moldova	Senegal	VaticanCity
Monaco	Serbia	Venezuela
Mongolia	Seychelles	Vietnam
Montenegro	SierraLeone	WesternSahara
Morocco	Singapore	Yemen
Mozambique	Slovakia	Zambia
Myanmar(Burma)	Slovenia	Zimbabwe

5.12 Passwords

Learn how user passwords are set and reset using SFAPI.

You can use the `User` entity to create users, set their initial passwords, and reset user passwords. You have the following options to set or reset passwords:

- Upsert the `<password>` field directly.
- Use the `defaultPasswordField` processing parameter to upsert a user entity.

All passwords follow the password policy defined in [Admin Center > Password & Login Policy Settings](#).

Note

If a user's password is newly created or updated by an API admin, they're required to reset the password upon login. If a user updates their own password using API, they're not required to reset the password upon login.

Related Information

[Optional Business Processing Parameters \[page 96\]](#)

6 SFAPI Error Messages

This section explains the cause, meaning, and solutions for SFAPI error messages in SAP SuccessFactors modules.

6.1 Calibration Module Error Messages

UNSUPPORTED_OPERATIONS Error Codes

Error Message	Cause	Solution
Delete is not supported operation in Calibration Domain! Insert is not supported operation in Calibration Domain! Upsert is not supported operation in Calibration Domain!	The operations Delete, Insert and Upsert are not supported in the Calibration Domain.	Make sure not to do these operations in the Calibration Domain.

NO_PERMISSIONS_EXCEPTION Error Code

Error Message	Cause	Solution
Insufficient permissions to fulfill request.	User does not have sufficient permission to fulfill the request.	Check whether user has sufficient permission. For example: In recruiting management domain, check whether recruiting feature is enabled and user have permission for SFAPI. To enable SFAPI web services, please contact Support. To grant permission to a user, please go to the Admin tools.
Calibration feature is not enabled.	Calibration feature is not enabled for this company. To enable SFAPI web services, please contact Support.	Make sure that calibration feature is enabled before taking this action.
Calibration external summary feature is not enabled.	Calibration external summary feature is not enabled for this company.	Make sure that calibration summary feature is enabled before taking this action. To enable SFAPI web services, please contact Support.

6.2 Common Module Error Messages

Error Code	Error Message	Cause	Solution
INVALID_FIELD_VALUE	{id} is invalid!	The given id value is invalid.	The given id value is invalid.

Error Code	Error Message	Cause	Solution
INVALID_FIELD_VALUE	SurrogateKey is null!	SurrogateKey is not provided.	Make sure to provided SurrogateKey.
INVALID_FIELD_VALUE	{surrogateKey} + should be started with {prefix}.	SurrogateKey is not started with the defined prefix.	Check the format of the given SurrogateKey. Make sure the SurrogateKey is started with the defined prefix.
UNSUPPORTED_OPERATIONS	Entity type '{fullEntityType}' doesn't support {feature} operation!	The operation is not supported in this entity type.	Check whether the operation is supported in this entity type.
NO_INPUT_PROVIDED	No SFObjct input provided.	No SFObjct input provided.	Make sure to input SFObjct
NO_PERMISSIONS_EXCEPTION	Insufficient permissions to fulfill request.	User does not have sufficient permission to fulfill the request.	<p>Check whether user has sufficient permission. For example:</p> <p>In recruiting management domain, check whether recruiting feature is enabled and user have permission for SFAPI.</p> <p>To enable SFAPI web services, please contact Support.</p> <p>To grant permission to a user, please go to the Admin tools.</p>
FAILED_FORMAT_DATE_VALUE	The date value/ ISO8601 date value is null or empty.	The date value/ ISO8601 date value is not provided.	Make sure the date value in the system is not empty.
SCHEMA_VALIDATION	Fault code: FAILED_XML_SCHEMA_VALIDATION Detail messages differ from cases. Sample: One of {"urn:sfobject.sfapi.successfactors.com":id,urn:sfobject.sfapi.successfactors.com":type}' is expected.	The input SOAP message is invalid.	Validate the XML document against the schema and make sure it is valid.
FAILED_PARSE_DATE_STRING	Failed to parse date string/ ISO8601 date string. The input date string is: {dateTimeStr}.	The given date value does not conform to the date format/ISO8061 standard.	Check the given date value and make sure it conforms to the standard.
FAILED_PARSE_DATE_STRING	The date string/ISO8601 date string is null or empty.	The date / ISO8601 date is not provided.	Make sure to provide the date.
INVALID_ENTITY_ID	The entity type is null or blank!	No entity type provided.	Make sure to Please provide the entity type.
INVALID_ENTITY_ID	The entity type {entityType} is invalid!	The given entity type is invalid.	Make sure to provide a valid entity type.
UNDEFINED_ENTITY_TYPE ID	Undefined entity type '{ entityType fullEntityType} '!	The given entity type is not defined in the entity model.	Make sure to provide a correct entity type.
UNDEFINED_ENTITY_TYPE ID	Entity type '{entityType}' is undefined!	The given entity type is not defined.	Make sure to provide a correct entity type.

Error Code	Error Message	Cause	Solution
UNDEFINED_ENTITY_TYPE ID	Entity type name '{entityType}' doesn't allow suffix!		Make sure to provide a valid entity type.
UNDEFINED_ENTITY_TYPE ID	Entity type name '{entityType}' doesn't allow sub entity!		Make sure to provide a valid entity type.
INVALID_SESSION	Invalid SFAPI session!	User did not login or the session is out of time.	Please log in first.
INVALID_SESSION	Invalid session detected due to certificate source check error!		
FAILED_AUTHENTICATION	Login failure due to the invalid company!	The given company id is invalid.	Make sure the given company id is correct.
FAILED_AUTHENTICATION	Authentication failed, password has expired.	Password has expired.	Please contact the admin to reset your password.
FAILED_AUTHENTICATION	Authentication failed, missing username, password, or clientIP.	Missing username, password, or losing clientIP.	Make sure to input username and password. If losing clientIP, please contact Support.
FAILED_AUTHENTICATION	Authentication failed, user account needs to be validated.	This user account needs to be validated.	Please contact the admin to validate this account.
FAILED_AUTHENTICATION	Authentication failed, user account is locked.	User account is locked.	Please contact the admin to reset this user's account.
FAILED_AUTHENTICATION	Authentication failed, invalid user id or password.	The given username or password is not correct.	Make sure to provide correct username and password.
FAILED_AUTHENTICATION	Authentication failed, no login permission.	User doesn't have the permission to log in to SFAPI or username is invalid.	Check whether the username is valid and make sure the user has the permission to access SFAPI.
FAILED_AUTHENTICATION	Authentication failed due to expired company.	Company has expired.	Contact Support to extend the company account.
FAILED_AUTHENTICATION	" Authentication failed. Attempted login from unauthorized ip: " + SFAPIW-SHelper.getClientIP(ctx) + " to company id: " + credential.getCompanyId() + " by username: " + credential.getUserName().	User is attempting to log in from an unauthorized IP.	Check whether user's current ip is authorized. To edit the ip for a user, please go to Admin Tools->Password Policy Settings.
FAILED_AUTHENTICATION	Authentication failed due to inactive company.	Company is inactive.	Please contact Support to enable company instance for SFAPI.
FAILED_AUTHENTICATION	USER_NO_LOGIN_PERMISSION	User doesn't have login permission	Assign required permissions. For more information, see Permissions [page 84]

Error Code	Error Message	Cause	Solution
FAILED_AUTHENTICATION	USER_NEEDS_VALIDATION	User account needs to be activated	Activate account. For more information, see Overview [page 84]
SOAPFACTORY	SOAPFactory Exception.	Failed to create SOAP fault internally.	Please contact Support.
QUERY_PARAMETER_MAX_ROW_EXCEEDS_LIMIT	parameter 'maxRows' with supplied value '{maxRows}' exceeds max limit '200'	The value of maxRows must not be greater than 200 .	Make sure the given value of maxRows is less than 200.
INVALID_LOCALE	'{value}' is not valid locale. Available locales are: en_US, zh_CN, zh_TW, de_DE	The given locale is not valid.	Please provide a correct locale, the available locales are listed in the error message.
INVALID_LOCALE	INVALID_LOCALE. Language Pack is not enabled.	Language Pack is not enabled for your company and the given default locale is not en_US.	Pack is not enabled, please provide the en_US for default locale field.
INVALID_LOCALE	INVALID_LOCALE. Invalid default locale:{value}	The given default locale is not valid.	Please provide correct default locale, the value should be one of the enabled Language Pack for your company.
INVALID_OPERATION_PARAMETER	Invalid '{param}' value: {value}	The given SFPParameter value is not valid.	Please provide a correct value for SFPParameter, make sure the given value corresponds to the SFPParameter type.
INVALID_OPERATION_PARAMETER	parameter '{param}' with supplied value '{value}' exceeds max limit '{limit}'	The given SFPParameter value exceeds the limit.	Please provide a correct value for SFPParameter, for the SFPParameter that expects an integer, please provide a integer value and make sure the value does not exceed the limit.
INVALID_QUERY_SESSION	Invalid query session id = {session-id}!	This error occurs during a rolling upgrade in the maintenance window when a user performs a QueryMore operation and the SF_API_SESSION_KEY_QUERYSESSION attribute for the session is larger than 2M.	Please wait for the rolling upgrade to finish and retry.

6.3 CDP Module Error Messages

CDP_ERROR_CERT_MISSING_FIELD CERT_FIELD_NOSUFFICIENT Error Code

Error Message	Cause	Solution
{field} required Sample: CertExpiration-Date required.	{field} is required and no value is provided for {field}.	Make sure to provide value for this field.

CDP_ERROR_TEMPLATE_NOT_FOUND CONFIG_ERROR Error Code

Error Message	Cause	Solution
The company does not have default LearningActivity template.	The company does not have default LearningActivity template.	Contact Support to add a new Learning Activity Plan template.

CDP_ERROR_NO_ACCESS_PERMISSION CDP_ACCESS_PERMISSION_ERROR Error Code

Error Message	Cause	Solution
User doesn't have CDP_ACCESS/Import Permission. User doesn't have permission to access CDP or do import.	Check whether the user has permission to access CDP or do import.	Grant the Administrator Permissions Manage Career Development Import User Relationship for Learning Administrator and Educational Representative permission.

CDP_ERROR_VALIDATION_FAILED VALUE_VALIDATION Error Code

Error Message	Cause	Solution
{fieldId} is the invalidate value in the template.	Given field id {fieldId} is the invalidate value in the template.	Check the field definition in the template and make sure to give a valid field id.
BeginDate must before EndDate.	The given begin date is not before end date.	Make sure the begin date is before end date.
You cannot set status to {status} because EndDate is after current time.	Learning activity start/end dates must before or equal to the day the learning activity is completed.	Make sure the EndDate and the status value are correct.
CertIssueDate must before CertExpirationDate.	CertIssueDate must be before the CertExpirationDate.	Make sure CertIssueDate and CertExpirationDate are correct.
Invalid user id for Assignee/ CompletedBy/ PlannedBy/ ConfirmedBy.	The given value of user id is invalid.	Make sure the given user id is correct for Assignee/ CompletedBy/ PlannedBy/ ConfirmedBy.

INVALID_QUERY_PARAMETER Error Code

Error Message	Cause	Solution
Invalid starting row: {startingRow}.	The given value of startingRow is invalid.	Make sure the given value of startingRow is an integer and is greater than 0.
Invalid starting row: {maxRow}.	The given value of maxRow is invalid.	Make sure the given value of maxRows is an integer and is between 0 and 200.

INVALID_REQUEST_MESSAGE Error Code

Error Message	Cause	Solution
Invalid request message! Error: Request record count of {sfObjects.size()} exceeds max batch size of 200.	Too many records (sfobjects) input.	Make sure to input no more than 200 sfobjects.
Request message has specified inconsistent entity type! Global entity type: {type}, entity type: {sfObject.getType()}. Occurred at row {rowIndex}.	The given entity type of this sfobject is not consistent with the global entity type.	Make sure that all the sfobjects are the same entity type.
For Update/Delete operation, request message is missing required Id field. Occurred at row {rowIndex}.	Id is not provided when performing Update/Delete operation.	Make sure to provide Id when performing Update/Delete operation.
Invalid Id field value ' sfObject.getId() ' in SFObject! Occurred at row {rowIndex}.	The given value of Id field is invalid.	Make sure to provide a valid value for Id field.
For Insert/Upsert operation, request message cannot specify Id field. Occurred at row {rowIndex}.	Id field is specified when performing Insert/Upsert operation.	Make sure not to specify Id field when performing Insert/Upsert operation.
Invalid id element is in the SFObject. Occurred at row {rowIndex}.	Invalid Id element is in the SFObject.	Check whether existing invalid Id element in the SFObject.
Found duplicated field name {local-Name}, current:= {e.getNodeName()}, previous:= {fieldNames.get(localName)}. Occurred at row {rowIndex}.	There is duplicated field name in the SFObject.	Check whether existing duplicated field name in the SFObject. One field name can only appear once.
Required field {f.getFieldName()} is missing in message#={fMessageNo}.	Missing required field {f.getFieldName()}.	Make sure to provide the required field {f.getFieldName()}.
Required field {f.getFieldName()} is null/empty in message#={fMessageNo}!	The given value of required field {f.getFieldName()} is null/empty.	Make sure to provide value for required field {f.getFieldName()}.
Undefined field {fFieldName} at {getLocationText()}!	The given field name is undefined in the entity model.	Check whether the field name is defined in the entity model. Use describe operation to list all supported field names.
Required field {fFieldName} cannot be null or empty at {getLocationText()}!	The given value of required field {fFieldName()} cannot be null/empty.	Make sure to provide value for required field {fFieldName()}.
Invalid foreign surrogate key {value} of field {fFieldName} at {getLocationText()}!	The given value of surrogate key is invalid.	Make sure to provide a valid value for surrogate key.
String length exceed the limit(actual={buffer.length}, limit={field.getMaxLength()}) at {getLocationText()}!	The length of given value (type: string) exceeds the limit of this field.	Make sure the length of value(type: string) is valid.
Invalid boolean value {value} (valid values "true", "false", "1", "0") at {getLocationText()}!	The given value (type: boolean) is invalid.	Make sure the value provided is in {"true", "false", "1", "0"}
Invalid integer/long/date/datetime/float/double value {value} at {getLocationText()}!	The given value is invalid for this field.	Make sure the value is valid for its type.

UNSUPPORTED_OPERATIONS Error Code

Error Message	Cause	Solution
Delete is not supported operation in Transcript Domain!	The operations Delete, Insert, Query, Querymore, and Update are not supported in the Transcript Domain.	Make sure not to do these operations in the Transcript Domain. The Transcript Domain only supports the Upsert operation!
Insert is not supported operation in Transcript Domain!		
Query is not supported operation in Transcript Domain!		
queryMore is not supported operation in Transcript Domain!		
Update is not supported operation in Transcript Domain!		

NO_PERMISSIONS_EXCEPTION Error Code

Error Message	Cause	Solution
Insufficient permissions to fulfill request.	User does not have sufficient permission to fulfill the request.	<p>Check whether the user has sufficient permission. For example:</p> <p>In recruiting management domain, check whether the recruiting feature is enabled and the user has permission for SFAPI.</p> <p>To enable SFAPI web services, please contact Support.</p> <p>To grant permission to a user, please go to Admin tools.</p>
PermissionListBean is null.	Failed to get permissionListBean from current session.	Maybe the session is out of time. Try to re-login. If you still see this error message, please contact Support.

CDP_ERROR_FEATURE_NOT_SUPPORTEDFEATURE_DISABLED Error Code

Error Message	Cause	Solution
Company doesn't have the Transcript feature.	Transcript feature is not enabled for this company.	Enable the Transcript feature first.

6.4 RCM Module Error Messages

INVALID_TEMPLATE_ID Error Code

Error Message	Cause	Solution
Template Id given as Invalid One.	The given template Id is invalid.	Make sure the given template Id is correct.

NO_PERMISSIONS_EXCEPTION Error Code

Error Message	Cause	Solution
Insufficient permissions to fulfill request.	User does not have sufficient permission to fulfill the request.	<p>Check whether the user has sufficient permission. For example:</p> <p>In recruiting management domain, check whether the recruiting feature is enabled and the user has permission for SFAPI.</p> <p>To enable SFAPI web services, please contact Support.</p> <p>To grant permission to a user, please go to Admin tools.</p>

6.5 User Module Error Messages

INVALID_FIELD_VALUE Error Code

Error Message	Cause	Solution
INVALID_FIELD_VALUE {fieldName} parse failure on value {value}.	The data type of the given value is not consistent with the defined type of this field in the entity model.	Use the correct data value for this field.
INVALID_FIELD_VALUE Field name: {name} Field value: {value}. {detail message}.	The given value is invalid for this field.	Make sure the given value is a valid value for this field.

INVALID_FIELD_VALUE Error Code

Error Message	Cause	Solution
INVALID_FIELD_TYPE {columnName} has an invalid type.	The data type defined in the entity model is invalid.	<p>Check whether the data type of this field is defined correctly in the entity model. The data type must be defined as one of the following:</p> <ul style="list-style-type: none">• double• float• int• long• boolean• date• datetime• string

USER_NOT_FOUND Error Code

Error Message	Cause	Solution
Cannot find user by user id. InternalId = {internalId}.	Cannot find user whose id is equal to the given id.	Make sure to use the correct user id.

Error Message	Cause	Solution
USER_NOT_FOUND Field name: managerid/ hrid Field value: {value} Cannot find user by user id. InternalId={value}.	Cannot find manager/hr by the given manager id/ hrid	Make sure to use the correct manager id or hr id.

INVALID_FIELD_NAME Error Code

Error Message	Cause	Solution
INVALID_FIELD_NAME {columnName}.	The given field name is not found in the entity model.	Check whether the field name is defined in the entity model. Use describe operation to list all supported field names.

INVALID_PARAMETER_VALUE Error Code

Error Message	Cause	Solution
Invalid value of default password parameter.	Given value of parameter "defaultPasswordField" is invalid.	Make sure the given value for parameter "defaultPasswordField" is one of "EXTERNALID", "USERNAME", "EMAIL", "FIRSTNAME", "LASTNAME".

NO_INPUT_PROVIDED Error Code

Error Message	Cause	Solution
No external id provided.	No external id is provided.	Make sure to provide external id.
No SFObjct fields provided.	No SFObjct is provided.	Make sure to provide SFObjct.

IMPORT_INITIALIZATION_EXCEPTION Error Code

Error Message	Cause	Solution
Unexpected error initializing Import engine.	Failed to initialize import engine internally.	Please contact Support.

UNSUPPORTED_OPERATIONS Error Code

Error Message	Cause	Solution
Delete is not supported operation in User Domain!	Delete is not a supported operation in the User Domain.	Make sure not to use the Delete operation in the User Domain.
Delete is not supported operation in Transcript Domain!	Delete is not a supported operation in the Transcript Domain.	Make sure not to use the Delete operation in the Transcript Domain.
Insert is not supported operation in Transcript Domain!	Insert is not a supported operation in the Transcript Domain.	Make sure not to use the Insert operation in the Transcript Domain.
Query is not supported operation in Transcript Domain!	Query is not a supported operation in the Transcript Domain.	Make sure not to use the Query operation in the Transcript Domain.
queryMore is not supported operation in Transcript Domain!	queryMore is not a supported operation in the Transcript Domain.	Make sure not to use the queryMore operation in the Transcript Domain.
Update is not supported operation in Transcript Domain!	Update is not a supported operation in the Transcript Domain.	Make sure not to use the Update operation in the Transcript Domain.

NO_PERMISSIONS_EXCEPTION Error Code

Error Message	Cause	Solution
Insufficient permissions to fulfill request.	User does not have sufficient permission to fulfill the request.	<p>Check whether user has sufficient permission. For example:</p> <p>In recruiting management domain, check whether the recruiting feature is enabled and the user has permission for SFAPI</p> <p>To enable SFAPI web services, please contact Support.</p> <p>To grant permission to a user, please go to Admin tools.</p>

UNDEFINED_ENTITY_TYPE Error Code

Error Message	Cause	Solution
Undefined entity type ' {fullEntity- Type} '!	The given entity type is not defined in the entity model.	Make sure to provide a correct entity type.

INVALID_USER_ID Error Code

Error Message	Cause	Solution
Failed to add/update user "{id} {full-Name}" ({department}; { division} ; {location}); Invalid User Id – "{id}".	Given use id is invalid.	Make sure to provide a valid user id.

INVALID_MANAGER_ID Error Code

Error Message	Cause	Solution
Failed to add/update user [{id}] "{full-Name}" ({department};{division};{location}) :Invalid Manager Id - "{managerId}".	Given manager id is invalid.	Make sure to provide a valid manager id.

INVALID_HR_ID Error Code

Error Message	Cause	Solution
Failed to add/update user[{id}] "{full-Name}" ({department};{division};{location}) :Invalid HR ID – "HRId".	Given hr id is invalid.	Make sure to provide a valid manager id.

MANAGER_CYCLE_DETECTED Error Code

Error Message	Cause	Solution
Failed to add/update user [{id}] "{full-Name}" ({department};{division};{location}) : Manager Cycle detected with this manager Id – "{managerId}"	Manager cycle detected.	Check the user's manager id and make sure not exist any manager cycle.

INVALID_USERNAME Error Code

Error Message	Cause	Solution
Failed to add/update user [{id}] "{fullName}" ({department};{division};{location}) :Invalid Username - "{userName}"	The given username is invalid.	Make sure to provide a valid username.

DUPLICATE_USERNAME Error Code

Error Message	Cause	Solution
Failed to add/update user [{id}] "{fullName}" ({department};{division};{location}) : Duplicate Username - "{userName}".	The given username has already existed in the system.	Check the username and make sure not to provide duplicate usernames.

PASSWORD_EXPIRED Error Code

Error	Cause	Solution
Password expired.	Password has expired.	Please contact the admin to reset your password.

USER_NOT_FOUND Error Code

Error	Cause	Solution
Failed to deactivate external user: "{id}" – User does not exist.	User not found by given id.	Make sure to provide a correct id.

ACCOUNT_LOCKED Error Code

Error	Cause	Solution
Account locked.	User account is locked.	Please contact the admin to reset this user's account.

REQUIRED_COLUMN_MISSING Error Code

Error	Cause	Solution
Error: Missing required field: {column} for user {firstName} {lastName} [{id}].	Missing field! Need to provide {column} field for this user.	Make sure to provide required field for every user.

INVALID_MATRIX_MANAGER_ID Error Code

Error	Cause	Solution
Failed to add/update user [{id}] "{fullName}" ({department};{division};{location}) : Invalid Matrix Manager List – "{matrixManagerList}".	The given matrix manager list has an invalid matrix manager id.	Make sure to provide valid matrix manager ids.

INVALID_CUSTOM_MANAGER_ID Error Code

Error	Cause	Solution
Failed to add/update user [{id}] "{fullName}" ({department};{division};{location}) : Invalid Custom Manager List – "{customManagerList}"	The given custom manager list has an invalid matrix manager id.	Make sure to provide valid custom manager ids.

FAILED_UPDATE_DIRECT_REPORT_COUNT_FOR_MGR Error Code

Error	Cause	Solution
Exception caught when resetting direct report counts for user "{id}".	Error occurred during updating direct report count for this user.	Please contact Support.

ACCOUNT_VALIDATION_NEEDED Error Code

Error	Cause	Solution
Account validation needed.	This user account needs to be validated.	Please contact the admin to validate this account.

USER_LIMIT_VIOLATION Error Code

Error	Cause	Solution
Failed to add/update user [{id}] "{fullName}" ({department};{division};{location}) : User License Limit is exceeded.	User license limit is exceeded.	Please contact Support to check the user license.

MAX_LEN_EXCEEDED_IN_A_FIELD Error Code

Error	Cause	Solution
Field {field} for user [{id}] is too big: max length is {maxLength} characters; given data is {length} characters long.	The given data's length exceeds the max length of this field.	Check the given data and make sure to keep the length of data is valid.

FAILED_TO_TRANSFER_MGR Error Code

Error	Cause	Solution
Failed to transfer user "{id}" from old mgr "{oldManagerId}" to new mgr "{newManagerId}".	Error occurred during performing manager transfer.	Please contact Support.

FAILED_TO_TRANSFER_MGR_FAILED_TO_RESTORE_OLDMGR Error Code

Error	Cause	Solution
Failed to change manager from "{newMangerId}" back to "{oldManagerId}" for user "{id}".	Error occurred during performing manager transfer.	Please contact Support.

FAILED_TO_REMOVE_FOLDERS_FOR_INACTIVE_USER Error Code

Error	Cause	Solution
Failed to remove folders for inactive user: "{id}".	User is inactive. Cannot remove folders for inactive user.	Check the user's status to make sure the user is active before removing the folder for this user.

FAILED_TO_VALIDATE_MGR Error Code

Error	Cause	Solution
Failed to update manager for user [{id}] ({tmpDeptInfo}): Invalid Manager Id - {managerId}.	The given manager id is invalid.	The given manager id is invalid.

FAILED_TO_VALIDATE_MGR_FAILED_TO_RESTORE_OLDMGR Error Code

Error	Cause	Solution
Failed to change manager from "{managerId}" back to "{oldManagerId}" for user {id} ({tmpDeptInfo}).	Error occurred during changing manager.	Please contact Support.

FAILED_TO_VALIDATE_MGR_INTERNAL_ERROR Error Code

Error	Cause	Solution
Exception caught when re-evaluating Manager "{managerId}".	Internal error!	Please contact Support.

FAILED_TO_VALIDATE_HR Error Code

Error	Cause	Solution
Failed to update HR for user [{"id }"] ("{tmpDeptInfo}"): Invalid HR Id - "{tmpHRId}".	Given hr id is invalid.	Make sure to provide a valid hr id.

FAILED_TO_VALIDATE_HR_INTERNAL_ERROR Error Code

Error	Cause	Solution
Exception caught when re-evaluating HR " + tmpHRId.	Internal error!	Please contact Support.

FEATURE_DISABLED Error Code

Error	Cause	Solution
Company instance not enabled for SFAPI!	Company instance not enabled for SFAPI!	Please contact Support to enable the company instance for SFAPI.

7 New DateTime Fields in Recruiting Entities

New fields of type DateTime are added to some of the Recruiting SFAPI entities to fix a bug in the existing Date fields.

New Fields of Type DateTime

Entity	Field Name (Label)
JobRequisition	jobCloseDateTime (jobCloseDateTime) dateTimeCreated (dateTimeCreated) startDateTime (as defined in requisition template for start date property) endDateTime (as defined in requisition template for end date property)
JobPosting	postStartDateTime (postStartDateTime) postEndDateTime (postEndDateTime) postExpirationDateTime (postExpirationDateTime)
JobApplication	lastModifiedForAppStatusDateTime (lastModifiedForAppStatusDateTime) hiredOnDateTime (hiredOnDateTime) exportedOnDateTime (exportedOnDateTime) applicationDateTime (as defined in the application template for application date property) reviewDateTime (as defined in the application template for review date property)
AssessmentOrder	createdDateTime (createdDateTime) lastModifiedDateDateTime (lastModifiedDateDateTime)

Note

We recommend you use either the Date or the new DateTime fields in the payload while updating the data. However, if you provide both Date and DateTime fields in the payload, the value from the last field is considered for the update. The DateTime fields must follow the Date fields to see the right dates.

In the following example, the value from applicationDateTime field is considered for the database update.

Sample Code

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:sfobject.sfapi.successfactors.com">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:update>
      <urn:type>JobApplication$792</urn:type>
      <urn:sfobject>
        <urn:type>JobApplication$792</urn:type>
        <urn:id>684</urn:id>
```

```
<urn:applicationDate>2020-10-14</urn:applicationDate>
<urn:applicationDateTime>2020-10-14T13:40:50Z</
urn:applicationDateTime>
  </urn:sfobject>
</urn:update>
</soapenv:Body>
</soapenv:Envelope>
```

8 SFAPI Best Practices

Read and follow the best practices in this document to ensure optimal SFAPI performance and better user experience.

Query only modified records

Instead of querying all records, query only the records that have been modified since your last execution for integration use cases, and use server pagination to ensure stable results.

Do not run jobs too often to get real-time results

Repeating queries in much less than one hour will consume too much API resource, which may be throttled in the future. In extreme cases, you may even be denied of service because of frequent queries, especially ones that require a lot of backend processing.

Avoid excessive queries for single records

Querying records one at a time does not take advantage of database optimization capabilities. Singleton queries are often used to perform in-memory joins and should be avoided whenever possible.

Use \$select to get the properties you need

Without the `$select` query option, a query returns all available properties of an entity. For better performance, we recommend that you add the `$select` system query option to specify only the properties you need.

Tune your batch requests into proper sizes

The OData API can return a maximum number of 1000 records in a single page. You should tune your batch sizes to be as large as possible. For more complex transactions, you may need to decrease the size to avoid HTTP timeouts.

Tune your client wait time to match the system wait time

Your client should be set to wait a reasonable amount of time before timing out. Complex operations can take as long as 10 minutes and our network and servers will continue to process a transaction for that long. It is best to wait for the transaction to complete rather than just wasting the transaction.

You can also tune your transaction to avoid timeouts.

Avoid excessive client multithreading

To prevent server overloads and ensure high availability, we recommend a maximum of 10 concurrent requests / threads per client. Multithreading is only allowed for editing single records. It should not be attempted for queries/read operations, massive upserts, or batch operations.

9 Change History

Learn about changes to the documentation for SAP SuccessFactors HCM suite SFAPI: Developer Guide in recent releases.

2H 2023

Type of Change	Description	More Info
January 19, 2024		
Added	We added a note about the deprecation of SOAP SFAPIs for Employee Central, except the <code>CompoundEmployee</code> API.	Deprecation of Partner API, SFAPI Ad-hoc, and SFAPI for Simple Entities [page 10]
October 14, 2023		
Changed	We moved Change History to the end of the guide.	Introduction [page 4]

1H 2022

Type of Change	Description	More Info
September 9, 2022		
Changed	The location of DC2/DC56 API server is corrected as Eemshaven, The Netherlands.	List of SAP SuccessFactors API Servers [page 5]

2H 2021



Type of Change	Description	More Info
Oct 8, 2021		
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 5]

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.

© 2024 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

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

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

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

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