## Overview

The SAP SF application is a collection of helper utilites, integration objects, and more. This document is meant to highlight the major components and capabilities.

# Helpful Resources

* SAP API Refence: <https://api.sap.com/package/SuccessFactorsEmployeeCentral/all>
* SAP OData Reference Guide: <https://help.sap.com/doc/7efdca36492e47c7b20ab92c4ca6323c/2211/en-US/SF_EC_OData_API_REF.pdf>

# Pre-Requisites

## Deploy Plug-ins

This utility requires the Date and Time Utilities plug-in from the App Market.

## API Key

To get started, you will need a SAP Success Factor environment and API Key.

If you want to try things out, you can connect with the SAP Sandbox @ <https://api.sap.com> and an API Key can be generated @ <https://api.sap.com/settings>



# Import

## Configure Import Properties

|  |  |  |
| --- | --- | --- |
| **Property** | **Value** | **Notes** |
| connectedSystem.\_a-0000e9b9-a17c-8000-d204-01ef9001ef90\_1107016.baseUrl | ***Your SF environment endpoint*** | Use <https://sandbox.api.sap.com/successfactors/odata/v2> to work against the SAP API Sandbox. |
| connectedSystem.\_a-0000e9b9-a17c-8000-d204-01ef9001ef90\_1107016.apiKeyValue | ***Your SF API Key*** | Use the API Key from your environment or use a value generated from the SAP API Sandbox. |

## Import Package

Import the package **SAP Success Factors with Groundswell v1.zip** and include the custom properties file noted above.

### Key Features

## Core Integrations

### rule!SAPWG\_SF\_EXT\_GET\_record (Integration)

This integration acts as a wrapper for making a GET call to an SF endpoint

|  |  |
| --- | --- |
| **Parameters** | **Description** |
| record | Name of API endpoint to fetch (required) |
| search |  (optional) |
| pagingInfo | Batch information to retrieve (optional – default is first 10 records) |
| fetchTotalCount | Determines if the call returns a count or data (optional, default ***false***) |
| selection | Comma-separated fields to select from the results (optional)Review the SF data model for more detail on options, but for **PerPerson**, you can include ***personIdExternal,perPersonUuid,createdOn,personalInfoNav/firstName*** to retrieve fields from this object, and also child object. See **cons!SAPWG\_SF\_TEXT\_RECORD\_PERSON\_DEFAULT\_SELECTION** for an example. |
| expand | Which child relationships (Nav objects) should be fetched, use in parallel (optional)Review the SF data model for more detail on options, but for **PerPerson**, you can include ***personalInfoNav*** to retrieve that child object. See **cons!SAPWG\_SF\_TEXT\_RECORD\_PERSON\_DEFAULT\_EXPANSION** for an example. |
| filter | ODATA formatted filters to limit the data returned (optional)**rule!SAPWG\_RULE\_Person\_simpleFilter** shows a basic way format a simple list of OData “equals” filters.  |
| format | Format of data to return (optional – default JSON) |

### rule!SAPWG\_SF\_EXT\_POST\_upsert (Integration)

This integration acts as a wrapper for making a GET call to an SF endpoint

|  |  |
| --- | --- |
| **Parameters** | **Description** |
| body | One or more JSON representations of SF objects to create, update, or delete (required) |
| isDelete | Flag add a delete operator to all included body objects. Only use this if you want all incoming objects to be marked for deletion. (optional – default ***false***) |

## Data Subset Management

### Rule!SAPWG\_SF\_RULE\_General\_generateDatasubset (Expression)

This wrapper takes in integration results (from the GET call above) for total count and/or data pages and formats them to a common datasubset.

|  |  |
| --- | --- |
| **Parameters** | **Description** |
| pagingInfo | Batch information to retrieve (optional – default is first 10 records) |
| fetchTotalCount | Determines if the call returns a count or data (optional, default false) |
| totalCountResult | Result of an integration call to Success Factors representing the total count (optional) |
| pageResult | Result of an integration call to Success Factors (optional) |
| identifierField | Name of body field that should act as the identifier in the datasubset |

## Example

This application includes multiple examples of usage, but looking at **rule!SAPWG\_SF\_RULE\_Employment\_generateDatasubset**:



## Timestamp Manipulation

The application includes several helper rules for dealing with SAP time formatting:

|  |  |
| --- | --- |
| **Rule** | **Use** |
| SAPWG\_RULE\_Helper\_convertDatetimeToISOTime | Converts an Appian datetime to an ISO-formated date string |
| SAPWG\_RULE\_Helper\_convertDatetimeToEpoch | Converts an Appian datetime to SAP Epoch time |
| SAPWG\_RULE\_Helper\_convertEpochToDatetime | Converts SAP Epoch time to an Appian datetime |

# Example Record

The **SAPWG PerPerson record** is a synched record pulling Peronal Information ([PerPerson and PerPersonRelationship](https://api.sap.com/api/ECPersonalInformation/resource)), and [EmpJob](https://api.sap.com/api/ECEmploymentInformation/resource) information.

Core data source expression is handled by **SAPWG\_SF\_PerPerson\_recordDataSource.**

Ad-hoc synching is handled by **SAPWG\_SF\_PerPerson\_recordSync**.

# Wizard Use Case

**rule!SAPWG\_SF\_PAGE\_exampleTwo\_userCreation** is a wizard that will drive a designer through the fundamentals of creating a User object in SAP/Appian. Its purpose is to be educational, with a detailed approach to each step; do not use this for a production project.