SugarCRM

Bi-directional data between SugarCRM and Syspro

# Request:

**SMCRM-34** ([link](https://gabriellawhite.atlassian.net/browse/SMCRM-34))  
We want to be able to take updates from the SugarCRM Accounts and push them into Syspro.

**SMCRM-35** ([link](https://gabriellawhite.atlassian.net/browse/SMCRM-35))  
We want to be able to create an Account in Syspro from a Lead within SugarCRM.

# Requirements:

## SugarCRM:

The following end points will be needed to leverage.

### Authority Token:

The following endpoint grabs the authority token for other method calls.

A screenshot of a computer

Description automatically generated

Payload contains the credentials for the user and setting to connect to the site. The response contains the access token that will be needed for the call, when that access expires, a refresh token and when the refresh token expires.

### GET Accounts

The endpoint <https://summerclassics.sugarondemand.com/rest/v11_7/Accounts> can be used to be able to get account data. Utilizing the following payload can get Account records by “date\_modified”:

A screenshot of a computer code

Description automatically generated

Json file for reference : GET\_Accounts\_1record.json

### GET Leads

The endpoint <https://summerclassics.sugardemand.com/rest/v11_7/Leads> can be used to query leads records.

Utilizing a similar payload as Accounts, we can filter records to pull.

Note: possible changes will be needed to know when a Lead record is ready to be pulled into Syspro.

Note: There is a possibility that Leads are converted in Sugar to Accounts. If this is so, we don’t have to know when a Lead needs to be pushed into Syspro.

## SQL

The following table is being used to track updates of customers from Syspro:  
 [PRODUCT\_INFO].[SugarCRM].[ArCustomer\_Ref]

The following procedures is currently taking updates/creations from Syspro to send to SugarCRM:

[PRODUCT\_INFO].[SugarCrm].[UpdateCustomerReferenceTable]

When taking update from SugarCRM, we will need to update the ArCustomer\_Ref table to prevent circular updates being made. The following process is proposed:

1. Take updates from Sugar
2. Compare to ArCustomer\_Ref
3. If different
   1. Update ArCustomer\_Ref
   2. Push change to Syspro

## Syspro

Syspro operates with business objects. For customer updates the following business object will be used:

ARSSCS – (Syspro InfoZone - [link](https://infozone.syspro.com/Support/Lists/BusinessObjectsRegistration/DispForm.aspx?ID=60))

The business object has a parameter, document, and output in XML format:

Parameters – ARXXCS.XML  
Document – ARSSCSDOC.XML  
Output – ARSSCSOUT.XML

# Design

Creation of a new Talend Job that will be triggered within the current Sugar job: Main\_SugarExport\_v2.

The objective of the job will be to query Accounts/Leads, compare with and update ArCustomer\_Ref if the update needs to be pushed to Syspro, and then package and send the Business Object ARSSCS to Syspro.