**ENGEL & VÖLKERS**

Syed Furqan Ahmed

Salesforce Implementation

Contents

[Tracking module 3](#_Toc63288665)

[Description 3](#_Toc63288666)

[Configuration 3](#_Toc63288667)

[Metadata 4](#_Toc63288668)

[Logging Module 4](#_Toc63288669)

[Description 4](#_Toc63288670)

[Metadata 5](#_Toc63288671)

[Solarisbank Module 5](#_Toc63288672)

[Sync 5](#_Toc63288673)

[Configuration 5](#_Toc63288674)

[Operations 6](#_Toc63288675)

[Bank Balance 6](#_Toc63288676)

[Change Card Status 6](#_Toc63288677)

[Transaction Details 6](#_Toc63288678)

[Metadata 6](#_Toc63288679)

[Subscription Module 6](#_Toc63288680)

[Configuration 7](#_Toc63288681)

[Operations 8](#_Toc63288682)

[Resend Confirmation Email 8](#_Toc63288683)

[Sync Members 8](#_Toc63288684)

[Metadata 8](#_Toc63288685)

[Files Module 8](#_Toc63288686)

[Configuration 8](#_Toc63288687)

[Metadata 8](#_Toc63288688)

[Knowledge 9](#_Toc63288689)

[Faq 9](#_Toc63288690)

[Solarisbank 9](#_Toc63288691)

[Salesforce Configuration 9](#_Toc63288692)

[Case Auto response 9](#_Toc63288693)

[DKIM 9](#_Toc63288694)

[Email-To-Case 9](#_Toc63288695)

[Domain Redirects 9](#_Toc63288696)

[Chat 10](#_Toc63288697)

[Chat Button 10](#_Toc63288698)

[Web\_Support 10](#_Toc63288699)

[Chat Configuration 10](#_Toc63288700)

[Web\_Support 10](#_Toc63288701)

[Embedded Service 10](#_Toc63288702)

[Web Support 10](#_Toc63288703)

# Tracking module

## Description

Captures record data changes and logs the changes to custom object **“History Tracker“**.

Event on record insert and update is emitted using custom event “**Tracking Events**”. This ensures that there is no impact on default execution flow of object.

Changes done by user defined in custom label “Track\_History\_Exclusion” are not tracked

Captured metadata

1. Record id
2. Record name
3. Field name
4. Object name
5. Old field value
6. New filed value
7. User who changed/viewed record

Along with tracking changes record views are also tracked by added aura component **“Log\_Usage”** on record page.

Visualize record changes by user and views of record using tab “**Usage Dashboard”**

## Configuration

Fields that are to be tracked can be configured via custom metadata **“History Tracker Configuration”**

* Fields: comma separated field api name who needs to be tracked
* Object: Api name of object
* Track Insert: check the flag to track new records when inserted
* Track Update: check the flat to track field value changes
* Track Changes: check the flag to enable tracking
* Track View: check the flat to record user views on record

Note: Track changes is master flag if it is false then tracking update/insert is turned off

For every object after update and after insert trigger is to be configured and in trigger following code snippet needs to be added along with metadata record configuration

    //tracking service

    if(!System.Label.Track\_History\_Exclusion.contains(Userinfo.getUserName())

    TrackHistory.publishChanges('object api name');

To track record view go to lighting record page of respective object and add **Log\_Usage** Component

## Metadata

1. Apex
   1. TrackHistory
   2. AccountTrigger
   3. BankAccountTrigger
   4. BookingTrigger
   5. CampaignTrigger
   6. CardTrigger
   7. CaseTrigger
   8. IdentificationTrigger
   9. LeadTrigger
   10. OpportuntiyTrigger
   11. ScheduledOrderTrigger
   12. SubscriptionTrigger
   13. TaxIdentificationTrigger
   14. TrackingEventsTrigger
2. Object
   1. History\_Tracker\_Configuration\_\_mdt
   2. History\_Tracker\_\_c
3. Event
   1. Tracking\_Events\_\_e
4. Aura Component
   1. Log Usage

# Logging Module

## Description

Logs exceptions that occurs in apex and api call made via apex in custom object **“Log”.**

Log service is sent via by emitting event “**Log Event”.**

Captured Metadata

1. User Id
2. Record Id
3. Object Name
4. Url: For Api Call
5. Response Code: For Api Call
6. Request Body: For Api Call
7. Stack Trace : For Exception
8. Class,Method,Component : For Exception

## Metadata

1. Apex
   1. LogService
2. Object
   1. Log
3. Event
   1. Log Event

# Solarisbank Module

## Sync

Sync following details from Solarisbank. Initiate sync from **Account** page

* User Details
* Mobile Number
* Bank Accounts
* Cards
* Identification Attempts

Sync following details from Solarisbank. Initiate sync from **Bank Account** page

* Bookings
* Reservations
* Standing Orders

Add users **Solarisbank id** in account **Solarisbank Id** field on account record and select the tenant where that details are in Production/Mobile/Sandbox

## Configuration

Add Solarisbank credentials and URLs in custom metadata “Solarisbank Api Config”

1. Client\_Id\_\_c :
2. Client\_Secret\_\_c :
3. Version\_\_c: specify which version of uri to be called
4. Auth\_Uri\_\_c: Solarisbank authentication URL
5. Host\_Url\_\_c: specify Solarisbank host Uri

## Operations

### Bank Balance

Check bank account balance from **bank account** record using **check balance** button

### Change Card Status

Change card status from **Card Bank Case Item**. Status can be only changed only if it is not blocked by Solarisbank or closed by Solaris.

Valid status changes are

1. Block
2. Unblock
3. Close: Once card is closed it cannot be making active

### Transaction Details

Check booking details from **Booking Bank Case Item.**

## Metadata

1. Apex
   1. SyncAccountAndSolarisbankController
   2. SolarisbankService
2. Component
   1. SyncAccountAndSolarisbank
   2. SyncBankAccountAndSolarisbank
   3. BankCaseService

# Subscription Module

Customers can subscribe to multiple products of our org. Currently only Newsletter is available. Subscription record are stored under “**Subscription**” Object related to account.

Subscription info that are captured are

1. Subscription Terms and conditions accepted
2. Subscription Status: - Subscribed/unsubscribed/awaiting confirmation/conformation resent
3. Reason for unsubscribing
4. Type: Newsletter
5. Source: website source from where request is originated

## Configuration

Currently subscription is only integrated on ev-smartmoney.com WordPress site via open api

POST <https://ev-smartmoney.secure.force.com/services/apexrest/generate/lead/>

Parameters

fields[email][value]: Email of user

fields[productInterested][value]: Product user is subscribed to

fields[websiteSource][value]: website source

**When user sends a request to api.**

New account is created with type as prospect, preferred language as German and subscription record under it with status as awaiting confirmation.

Process builder “**Subscription Emails**” sends email to customer based on preferred language select one email templates from “**Subscription Emails**” classic email template folder. Email is sent for verification for subscription and thank you for successful subscription

**When user click verify button from email.**

he is redirected to community page

<https://support.ev-smartmoney.com/s/verifyemailaccount?language=en_US&product=Newsletter&email=useremail@domain.com>

where customer enter his name, phone and accepts terms and conditions as soon as he opens the page, customers subscription record is updated to subscribed and respective user details are updated on account

When customer is successfully subscribed to product, if campaign is running with the same product then he is added to campaign members list

## Operations

### Resend Confirmation Email

On subscription record if status is **awaiting confirmation** or c**onformation resent.** Then agent can explicitly send the mail to customer to verify and it also shows when the last email was sent

### Sync Members

On campaign record based on value that is selected on product sync all the contacts that are subscribed to same product. Delete the member whose subscription status is unsubscribe and add new members whose status is subscribed

## Metadata

1. Apex
   1. webLeadRestService
   2. SubscriptionService
   3. SyncCampaignMembersController
2. Aura Component
   1. SubscriptionResendEmail
   2. SyncCampaignMebmbers

# Files Module

To attach a file on any record custom button, attach file is to be used.

Using the component file can be tagged with pre-configured tags or add custom tag.

On selection of tag it shows all the existing files with same tag and new version of that file can be uploaded

## Configuration

Using custom metadata “**File Type Option**” default options is configured.

And for each object and record type specific options can also be configured

**Add button on object**

Create new lighting action button and select **“UploadFileAction”** as lighting component, add the button to page layout to start using the button

## Metadata

1. Apex
   1. UploadFileController
2. AuraComponent
   1. UploadFileAction

# Knowledge

Knowledge is currently configured to support 2 language English and German

## Faq

Faq record type knowledge is used to add public knowledge base that is available to customer

## Solarisbank

Solarisbank record type knowledge is used to add Solarisbank specific knowledge that is avilabel only to internal users that helps them during any process related to solarisbank

# Salesforce Configuration

## Case Auto response

When new case is created via web,phone,chat auto response email is trigger sending mail from “**Case email template**” folder

## DKIM

DKIM is setup for ev-smartmoney and weisshausinvestment domain in salesforce and cname for the following is registered in aws domain

## Email-To-Case

All emails that are sent to following emails are registered as general question record type case in salesforce

* [salesforce@ev-smartmoney.com](mailto:salesforce@ev-smartmoney.com)
* [support@ev-smartmoney.com](mailto:support@ev-smartmoney.com)
* [salesforce@weisshausinvestment.com](mailto:salesforce@weisshausinvestment.com)
* [support@weisshausinvestment.com](mailto:support@weisshausinvestment.com)

## Domain Redirects

Domain [support.ev-smartmoney.com](javascript:srcUp(%27%2F0I45I0000004CcA%3FretURL%3D%252F0I4%253FretURL%253D%25252Fsetup%25252Fhome%2526appLayout%253Dsetup%2526tour%253D%2526isdtp%253Dp1%2526sfdcIFrameOrigin%253Dhttps%25253A%25252F%25252Fev-smartmoney.lightning.force.com%2526sfdcIFrameHost%253Dweb%2526nonce%253D3ae2fe4d498efe4dddc7a5480449e918266ff9ffea2a3c1691fa1bc7bf301657%2526ltn_app_id%253D06m5I000000QN07QAG%2526clc%253D1%26isdtp%3Dp1%27);) is registered in salesforce that is linked with support community for redirection.

Certificate for domain is generated in salesforce as ca-signed certificate and signed using gandi.com.

After domain certificate is uploaded in salesforce the cname of subdomain is registered in aws

# Chat

## Chat Button

### Web\_Support

Web support chat button is configured with customized chat ui via vf page **"LiveAgentChatWindow”**

## Chat Configuration

### Web\_Support

This is configured for system admin and business development admin profile.

This is available to user with web support skill

There is no auto greeting message configured

## Embedded Service

### Web Support

This is default case creation embedded service button.

This is used in worpress site, qlpix site and salesforce support community