**ENGEL & VÖLKERS**

Syed Furqan Ahmed

Salesforce Implementation

Contents

[Tracking module 4](#_Toc100505339)

[Description 4](#_Toc100505340)

[Configuration 4](#_Toc100505341)

[Metadata 5](#_Toc100505342)

[Logging Module 5](#_Toc100505343)

[Description 5](#_Toc100505344)

[Metadata 6](#_Toc100505345)

[Solarisbank Module 6](#_Toc100505346)

[Configuration 6](#_Toc100505347)

[Operations 6](#_Toc100505348)

[New – Bank Case Item – Override 6](#_Toc100505349)

[Sync Solarisbank – Account Object 6](#_Toc100505350)

[Sync Auth0 – Account Object 7](#_Toc100505351)

[Solarisbank Actions – Account Object 7](#_Toc100505352)

[Sync Identification – KYC Details 7](#_Toc100505353)

[Auth0 Actions – Bank 7](#_Toc100505354)

[Bank Balance – Bank Account 7](#_Toc100505355)

[Sync Solarisbank – Bank Account 7](#_Toc100505356)

[Generate Tax Exception Order Link – Bank Account 8](#_Toc100505357)

[Generate Early Termination Order Link 8](#_Toc100505358)

[Update Person Details – Bank Case Item – Account Record Type 8](#_Toc100505359)

[New Person Details – Bank Case Item – Account Record Type 8](#_Toc100505360)

[Check Balance – Bank Case Item – Bank Account 8](#_Toc100505361)

[Request New Card – Bank Case Item – Bank Account 8](#_Toc100505362)

[Change Card Status – Bank Case Item - Card 8](#_Toc100505363)

[Transaction Details - Bank Case Item - Booking 8](#_Toc100505364)

[Verify User Identity – Case – Banking Case 8](#_Toc100505365)

[Start Jira ticket – Case – Banking Case 9](#_Toc100505366)

[Process Bug Info – Case – E&V Konto Case 9](#_Toc100505367)

[Metadata 9](#_Toc100505368)

[Subscription Module 10](#_Toc100505369)

[Configuration 10](#_Toc100505370)

[Operations 11](#_Toc100505371)

[Resend Confirmation Email 11](#_Toc100505372)

[Sync Members 11](#_Toc100505373)

[Metadata 11](#_Toc100505374)

[Files Module 12](#_Toc100505375)

[Configuration 12](#_Toc100505376)

[Metadata 12](#_Toc100505377)

[Knowledge 12](#_Toc100505378)

[Faq 12](#_Toc100505379)

[Solarisbank 12](#_Toc100505380)

[Salesforce Configuration 13](#_Toc100505381)

[Case Auto response 13](#_Toc100505382)

[DKIM 13](#_Toc100505383)

[Email-To-Case 13](#_Toc100505384)

[Domain Redirects 13](#_Toc100505385)

[Chat 13](#_Toc100505386)

[Chat Button 13](#_Toc100505387)

[Web\_Support 13](#_Toc100505388)

[Chat Configuration 13](#_Toc100505389)

[Web\_Support 13](#_Toc100505390)

[Embedded Service 14](#_Toc100505391)

[Web Support 14](#_Toc100505392)

[Integration 14](#_Toc100505393)

[Inbound 14](#_Toc100505394)

[Person Customer Account Create / Update 14](#_Toc100505395)

[Person Investment Account Create / Update 14](#_Toc100505396)

[Person Account Update 14](#_Toc100505397)

[Seizure Create 14](#_Toc100505398)

[Opportunity - Create 14](#_Toc100505399)

[Opportunity – Update 14](#_Toc100505400)

[Real Estate Asset 14](#_Toc100505401)

[Document Record Creation 15](#_Toc100505402)

[KYC Details Creation 15](#_Toc100505403)

[Outbound 15](#_Toc100505404)

[Document Read Request 15](#_Toc100505405)

[Product Create/Update 15](#_Toc100505406)

[Investment Opportunity Content Version Creation 15](#_Toc100505407)

[Investment Opportunity Details 15](#_Toc100505408)

[Community / Digital Sites 16](#_Toc100505409)

[Support 16](#_Toc100505410)

[Pages 16](#_Toc100505411)

[Marketing 16](#_Toc100505412)

[Pages 16](#_Toc100505413)

# 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 platform event “**Tracking Events**” , from trigger. 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 occur in apex and api call made via apex in custom object **“Log”.**

Log service is sent via by platform 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

## 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

### New – Bank Case Item – Override

New button of bank case item is overridden with component that allows agent a details step by step flow to select appropriate bank details and create bank case item records

### Sync Solarisbank – Account Object

Fetch data from Solarisbank based on Solarisbank id on account record and inserts record in

1. Bank – Logs all user details
2. Tax Identification – Logs all user tax details
3. Seizures – Logs seizures of user
4. KYC Details – Logs all kyc details of user
5. Attempts – Logs all attempts of kyc detail successfull
6. Bank Account – logs all bank account of user
7. Card – logs all cards of user

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 those details are in Production/Mobile/Sandbox

### Sync Auth0 – Account Object

Fetch user details from auth0 based on Auth0 id on user record

Sync account details

### Solarisbank Actions – Account Object

This component is only available to admin user, this allows admin to

1. Update Person Details
   1. Updates user details
2. New Person Identification
   1. Creates new identification attempt for the user

### Sync Identification – KYC Details

Fetch Identification details and attempts of it, inserts records in kyc details record if successful identification its attempts are fetched and inserted in attempt object record

### Auth0 Actions – Bank

This is available only on bank object for admin users, this allows users to toggle term deposit flag of user

### Bank Balance – Bank Account

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

### Sync Solarisbank – Bank Account

Fetches bank account related details form Solarisbank and inserts records in to

1. Bookings
2. Scheduled Order
3. Reservations

### Generate Tax Exception Order Link – Bank Account

For bank account type term deposit, this button generates a link to exception order form link of support community using which customer can register banking case

### Generate Early Termination Order Link

For bank account type term deposit, this button generates a link to termination order form link of support community using which customer can register banking case

### Update Person Details – Bank Case Item – Account Record Type

Allows agent to change customers user details , this steps sends one time verification code to customers registered number , which is needed to confirm user data change

### New Person Details – Bank Case Item – Account Record Type

Allows agent to create new identification attempt for customer

### Check Balance – Bank Case Item – Bank Account

Allows agent to check bank account balance of customer of associated bank account to this record

### Request New Card – Bank Case Item – Bank Account

Allows agent to request new card for customer

### Change Card Status – Bank Case Item - Card

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 - Bank Case Item - Booking

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

### Verify User Identity – Case – Banking Case

**Displays to agent user details that needs to be confirmed, on successful verification inserts records to case verification**

### Start Jira ticket – Case – Banking Case

Allows agent to create jira description based on case type and sub type, it also fetches relevant knowledge article from database to show what process needs to be followed,

Allows agent to attach any file and then sends mail to Solarisbank email address

### Process Bug Info – Case – E&V Konto Case

This allows agent to process case information and send email to Digibank in there required format to report but in app

E&V Konto case can be crated from https://support.ev-smartmoney.com/s/ev-konto-bug

## Metadata

1. Apex
   1. SyncAccountAndSolarisbankController
   2. SolarisbankService
2. Component
   1. SyncAccountAndSolarisbank
   2. SyncBankAccountAndSolarisbank
   3. BankCaseService
   4. AccountSolarisbankAdminPanal
   5. AccountRequestNewIdnow
   6. BankAccountEarlyTerminationLink
   7. BankAccountGetAccountBalance
   8. BankAccountTaxExemtionLink
   9. BankCaseCardStatusChange
   10. BankCaseCreateNewIdentifiactionAttempt
   11. BankCaseGetAccountBalance
   12. BankCaseGetBookingTransaction
   13. BankCaseNewOverride
   14. BankCaseRequestNewCard
   15. BankCaseTimedOrderCancle
   16. BankCaseUpdatePersonDetails
   17. BaseCaseVerifyAccount
   18. BookingsCaseView
   19. CardsCaseView
   20. CaseCreateBankCaseItem
   21. CaseJiraButton
   22. IdentificationCaseView
   23. IdentificationFetchInformation
   24. ReservationCaseView
   25. SolarisbankAdminPanal
   26. StandingOrderCaseView
   27. StatmentsCaseView
   28. SyncAccountAndAuth0
   29. SyncAccountAndSolarisbank
   30. SyncBankAccountAndSolarisbank

# Subscription Module

Customers can subscribe to multiple products of our org. 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: Type of production
5. Source: website source from where request is originated

## Configuration

#### Option 1

Using rest 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.

Flow “**Subscription Emails**” sends email to customer, email templates are stored in “**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

#### Option 1

Using marketing community website

Customer can register for product by providing email address on marketing website

Investment.ev-smartmoney.com

It creates a account and subscription record,

Email is sent to customer for verification of record

On verification subscription status is changed to subscibed

## 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
3. LWC
   1. Unsubscribe
   2. waitingList\_unsubscribe
   3. waitingList\_confirmEmail
   4. verifyEmailAccount

# 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 available 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

# Integration

## Inbound

### Person Customer Account Create / Update

* 1. Patch - /services/apexrest/Account/Create/
  2. Create a Person Customer Account based on autho id and person email

### Person Investment Account Create / Update

* 1. Patch - services/apexrest/Account/SMI/Create/
  2. Create person investment account based on evsm auth0 id and person email
  3. Creates investment newsletter subscription if marketing opt in is true

### Person Account Update

* 1. Patch - /services/data/v52.0/sobjects/Auth0\_Id\_\_c
     1. Updates user details
  2. Patch - /services/data/v52.0/sobjects/EVSM\_Auth0\_Id\_\_c
     1. Updates user details

### Seizure Create

* 1. POST - /services/data/v52.0/sobjects/Seizure\_\_c
  2. Backend team create a seizure record in salesforce when they receive any from solaribank

### Opportunity - Create

* 1. POST - services/data/v52.0/sobjects/Opportunity
  2. Backend team creates opportunity of different record type based on user selection

### Opportunity – Update

* 1. POST - services/data/v52.0/sobjects/Opportunity/External\_Id\_\_c
  2. Backend team updates opportunity using external key

### Real Estate Asset

* 1. GET - services/apexrest/Asset/RealEstate
     1. Returns real estate details based on external id asset id
  2. DELETE - services/apexrest/Asset/RealEstate
     1. Update is deleted by customer flag on delete request
  3. POST - services/apexrest/Asset/RealEstate
     1. Create real estate asset and evaluation records
  4. PUT - services/apexrest/Asset/RealEstate
     1. Updates real estate asset and evaluation records

### Document Record Creation

* 1. POST - services/data/v52.0/sobjects/Document\_\_c
  2. Creates a document record this record can be associated
     1. Account
     2. Opprtuntiy

### KYC Details Creation

* 1. POST - services/data/v52.0/sobjects/KYC\_Details\_\_c
  2. Creates kyc details record

## Outbound

### Document Read Request

* 1. GET - customers/:auth0Id/documents/:TicketId
  2. Request backend to generate a temp link to access document

### Product Create/Update

* 1. POST - /products
     1. Sends product details to backend when product is created
  2. PATCH - /products/:id
     1. Send updated product details to backend when product is updated

### Investment Opportunity Content Version Creation

* 1. POST - investments/:ExternalId/documents
  2. On attach of file from investment opportunity
  3. file details are sent to backend
  4. Intern backend deletes file from saleforce and creates new document record

### Investment Opportunity Details

* 1. POST - investments/:ExternalId/money-transferred
     1. Sends money transferred flag true to backend
  2. POST - investments/:ExternalId
     1. When investment status changes
     2. Corresponding change is sent to backend

# Community / Digital Sites

## Support

Support site is used to provide support to customer requests

### Pages

#### /agent-referral

Allows customer / partner to register and get reference code

#### /early-termination-form

Allows customer to register case with salesforce for early termination of account

#### /ev-konto-bug

Allows agents/customer to report bug in EV Konto app by creating case

#### /term-deposit-tax-exemption-order

Allows customer to register case with salesforce for term deposit of account

## Marketing

Marketing community used by marketing team to create campaigns

### Pages

#### /investment-warteliste

Allows customer to register for investment newsletter