

Implementation guide on ABAP SDK for Azure

|  |  |  |  |
| --- | --- | --- | --- |
| **Version No.** | **Date** | **Name** | **Description of change** |
| 1.0 | MAY, 2018 | ABAP SDK for Azure | Initial version |
|  |  |  |  |

Table of Contents

[1. Introduction 3](#_Toc514076343)

[2. Prerequisite 3](#_Toc514076344)

[3. Type of interface in Azure 3](#_Toc514076345)

[4. Steps for sending data to Azure EventHub 3](#_Toc514076346)

[4.1 Remote function call Destination Creation 4](#_Toc514076347)

[4.2 SAP-Azure Data Framework Configurations 4](#_Toc514076348)

[4.3 Usage of ABAP SDK for Azure EventHub 6](#_Toc514076349)

[5. Dos and Don’ts on usage of ABAP SDK 8](#_Toc514076350)

[6. Future Modification on implementation Guide 9](#_Toc514076351)

# Introduction

Intent of this document is to provide overview on the implementation of ABAP SDK for Azure ‘**EventHub**’ service. Azure Event Hubs is a highly scalable data ingress service that ingests millions of events per second so that you can process and analyze the massive amounts of data produced by your connected devices and applications. Once data is collected into an Event Hub, it can be transformed and stored using any real-time analytics provider or batching/storage adapters.

# Prerequisite

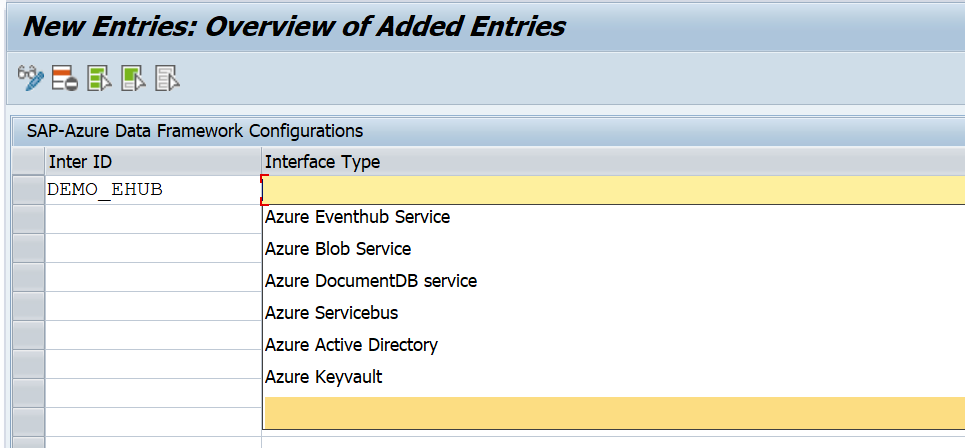
Installation of ABAP-SDK for Azure into your SAP system is required following the instruction of

[Installation guide of ABAP SDK for Azure](Link), for sending data to Azure EventHub.

# Type of interface in Azure

You can communicate to below Azure services through ABAP SDK.

* EVENTHUB - Azure EventHub Service
* BLOB - Azure Blob Service
* Document DB - service
* SERVICEBUS - Azure Servicebus
* AAD - Azure Active Directory
* KV- Azure Key vault



As part of initial version of implementation guide, follow the below steps to connect Azure EVENTHUB Service.

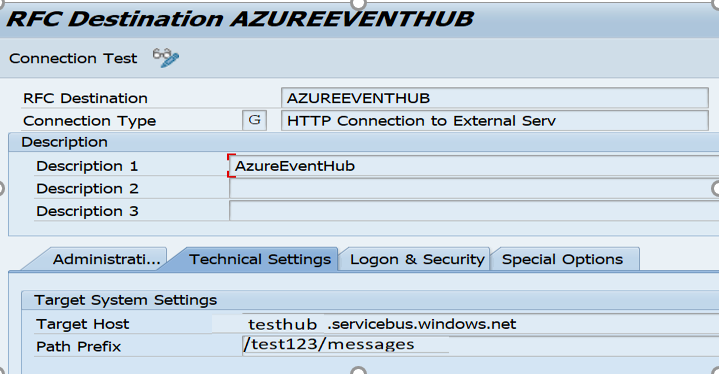
1. Steps for sending data to Azure EventHub

## 4.1 Remote function call Destination Creation

Create RFC destination in SAP (using transaction code SM59, Type ‘G’) as ‘HTTP connection to external server’ using the target host which you would receive from Azure Service team on getting subscription.

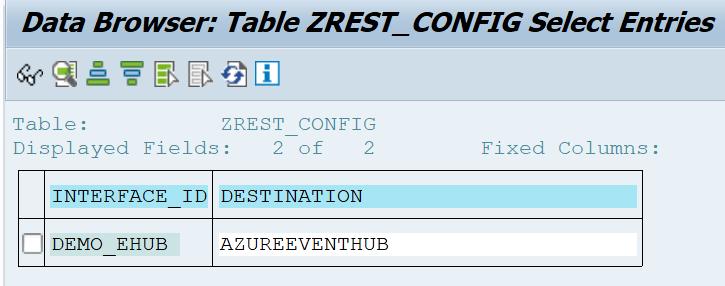
Details of configuration:

* RFC destination: AZUREEVENTHUB.
* Target Host: testhub.servicesbus.windows.net
* Path Prefix: /test123/messages



## SAP-Azure Data Framework Configurations

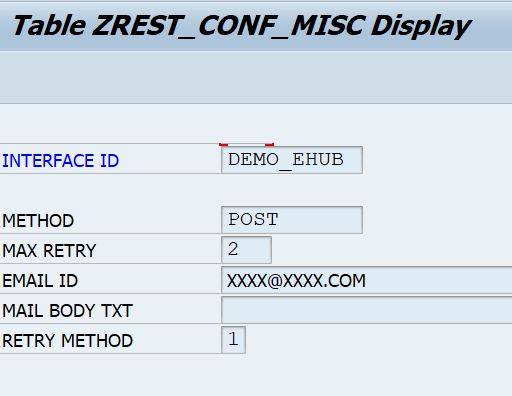
Create an interface Id ‘DEMO\_HUB’ and maintain RFC destination ‘AZUREEVENTHUB’ in table ‘**ZREST\_CONFIG’** (Configure Interface and RFC Destination) as shown below.



Create an entry in table ‘**ZREST\_CONF\_MISC’** for the above interface Id ‘DEMO\_EHUB’.

Details of configuration:

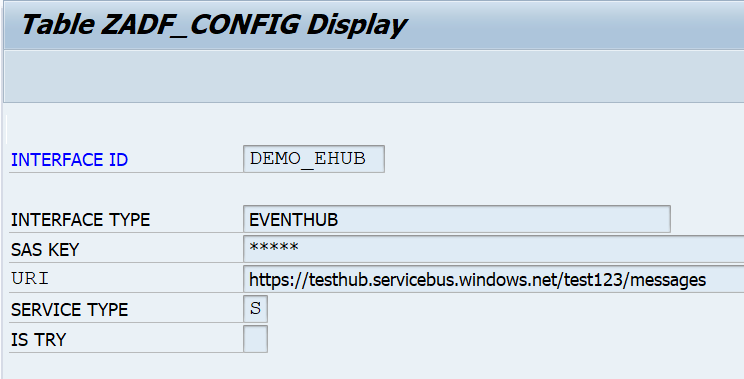
* METHOD is ‘POST’.
* MAX\_RETRY is number of retry in case of service failure.
* EMAIL\_ID is the email id for sending alerts.
* MAIL\_BODY\_TXT is Text Id to be maintained for the mail content.
* RETRY\_METHOD is type of retrial (Regular ‘0’ or exponential ‘1’).



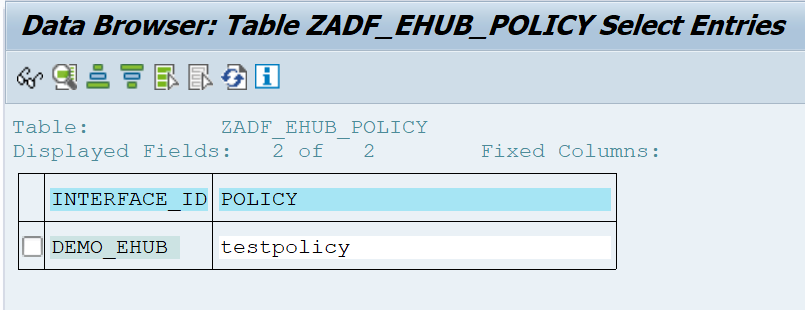
Create an entry in table ‘**ZADF\_CONFIG’** for the above interface Id ‘DEMO\_EHUB’.

Details of configuration:

* INTERFACE\_TYPE is ‘EVENTHUB’.
* SAS\_KEY is the shared key which you will receive from Azure team.
* URI is RFC destination Target host including path prefix of Azure EventHub to receive your data,
* SERVICE\_TYPE can be synchronous(S) or asynchronous(A), maintain ‘S’ for synchronous communication.
* IS\_TRY is a reprocessing flag, maintain as blank. However, it can be configured as ‘X’ for reprocessing in case of failure of services.



Create an entry in table ‘**ZADF\_EHUB\_POLICY’** for the interface Id ‘DEMO\_EHUB’ to maintain policy ‘testpolicy’ assigned in Azure for the URI location. You can get policy namespace from Azure team.



# 4.3 Usage of ABAP SDK for Azure EventHub

## 

\*&---------------------------------------------------------------------\*  
\*& Report  ZDEMO\_SEND\_TO\_EVENTHUB  
\*& Sample program for using ABAP SDK to connect Azure Eventhub  
\*&---------------------------------------------------------------------\*  
REPORT ZDEMO\_SEND\_TO\_EVENTHUB.

CONSTANTS: gc\_interface TYPE zinterface\_id VALUE 'DEMO\_EHUB'.  
TYPES: BEGIN OF lty\_data,  
          carrid    TYPE     s\_carr\_id,  
          connid    TYPE    s\_conn\_id,  
          fldate    TYPE    s\_date,  
          planetype TYPE    s\_planetye,  
          END OF lty\_data.  
  
DATA:       it\_headers        TYPE tihttpnvp,  
            wa\_headers        TYPE LINE OF tihttpnvp,  
            lv\_string         TYPE string,  
            lv\_response       TYPE string,  
            cx\_interface      TYPE REF TO zcx\_interace\_config\_missing,  
            cx\_http           TYPE REF TO zcx\_http\_client\_failed,  
            cx\_adf\_service    TYPE REF TO zcx\_adf\_service,  
            oref\_eventhub     TYPE REF TO zcl\_adf\_service\_eventhub,  
            oref              TYPE REF TO zcl\_adf\_service,  
            filter            TYPE zbusinessid,  
            lv\_http\_status    TYPE i,  
            lo\_json           TYPE REF TO cl\_trex\_json\_serializer,  
            lv1\_string        TYPE string,  
            lv\_xstring        TYPE xstring,  
            it\_data           TYPE STANDARD TABLE OF lty\_data.  
  
*\*Sample data population for sending it to Azure eventhub*  
SELECT  
   carrid  
   connid  
   fldate  
   planetype  
 FROM sflight  
 UP TO 10 ROWS  
 INTO TABLE it\_data.  
IF sy-subrc EQ 0.  
  
  TRY.  
*\*\*Calling Factory method to instantiate eventhub client*  
  
      oref = zcl\_adf\_service\_factory=>create( iv\_interface\_id = gc\_interface  
                                                         iv\_business\_identifier = filter ).  
      oref\_eventhub ?= oref.  
  
*\*\*Setting Expiry time*  
      CALL METHOD oref\_eventhub->add\_expiry\_time  
        EXPORTING  
          iv\_expiry\_hour = 0  
          iv\_expiry\_min  = 15  
          iv\_expiry\_sec  = 0.  
  
      wa\_headers-name = 'path\_prefix'.  
      wa\_headers-value = '/test123/messages'.  
      APPEND wa\_headers TO it\_headers.  
      CLEAR  wa\_headers.  
  
      CREATE OBJECT lo\_json  
        EXPORTING  
          data = it\_data.  
      lo\_json->serialize( ).  
      lv1\_string  = lo\_json->get\_data( ).  
  
  
*\*Convert input string data to Xstring format*  
      CALL FUNCTION 'SCMS\_STRING\_TO\_XSTRING'  
        EXPORTING  
          text   = lv1\_string  
        IMPORTING  
          buffer = lv\_xstring  
        EXCEPTIONS  
          failed = 1  
          OTHERS = 2.  
      IF sy-subrc <> 0.  
      ENDIF.  
*\*\*Sending Converted SAP data to Azure Eventhub*  
      CALL METHOD oref\_eventhub->send  
        EXPORTING  
          request        = lv\_xstring  *"Input XSTRING of SAP Business Event data*  
          it\_headers     = it\_headers  *"Header attributes*  
        IMPORTING  
          response       = lv\_response       *"Response from EventHub*  
          ev\_http\_status = lv\_http\_status.   *"Status*  
    CATCH zcx\_interace\_config\_missing INTO cx\_interface.  
      lv\_string = cx\_interface->get\_text( ).  
      MESSAGE lv\_string TYPE 'E'.  
    CATCH zcx\_http\_client\_failed INTO cx\_http .  
      lv\_string = cx\_http->get\_text( ).  
      MESSAGE lv\_string TYPE 'E'.  
    CATCH zcx\_adf\_service INTO cx\_adf\_service.  
      lv\_string = cx\_adf\_service->get\_text( ).  
      MESSAGE lv\_string TYPE 'E'.  
  ENDTRY.  
  IF lv\_http\_status NE '201' AND  
     lv\_http\_status NE '200'.  
    MESSAGE 'SAP Event data not sent to EventHub client' TYPE 'E'.  
  ELSE.  
    MESSAGE 'SAP Event data sent to EventHub client' TYPE 'S'.  
  ENDIF.  
  
ELSE.  
  MESSAGE 'No data found' TYPE 'E'.  
ENDIF.  *" If data present.*

# Dos and Don’ts on usage of ABAP SDK

ABAP SDK is primarily designed for real time eventing data from SAP to Azure EventHub as per initial version. This shouldn’t be used for huge volume of data in a batch processing as several parallel processing approaches are required explicitly to handle the bulk volume of data processing which has not been considered in the current version of ABAP SDK library.

# Future Modification on implementation Guide

This implementation guide has shown the steps for implementing Interface to Azure Event Hub, In successive versions we will publish guide on connecting to all other Azure Services like Blob, Servicebus, Cosmos DB, Azure Active directory and Azure key vault.