ABAP SDK Implementation guide for Azure Active Directory

https://github.com/Microsoft/ABAP-SDK-for-Azure

Author: Microsoft SAP Team

Version: 1.0

Contents

What is Azure Active Directory?

Azure Active Directory (Azure AD) provides an easy way for businesses to manage identity and access, both in the cloud and on-premises. Your users can use the same work or school account for single sign-on to any cloud and on-premises web application. Users can use their favorite devices, including iOS, Mac OS X, Android, and Windows. An Organization can protect sensitive data and applications both on-premises and in the cloud with integrated multi-factor authentication ensuring secure local and remote access. Azure AD extends your on-premises directories so that information workers can use a single organizational account to securely and consistently access their corporate resources. Azure AD also offers comprehensive reports, analytics, and self-service capabilities to reduce costs and enhance security. The Azure AD SLA ensures that your business always runs smoothly and can be scaled to enterprise levels.

For more details on Azure Active directory, visit Microsoft Azure Active Directory

Prerequisites

Make sure you have installed ABAP SDK for Azure in your SAP system. Refer document 'ABAP SDK for Azure – GitHub' for more details, Visit https://github.com/Microsoft/ABAP-SDK-for-Azure

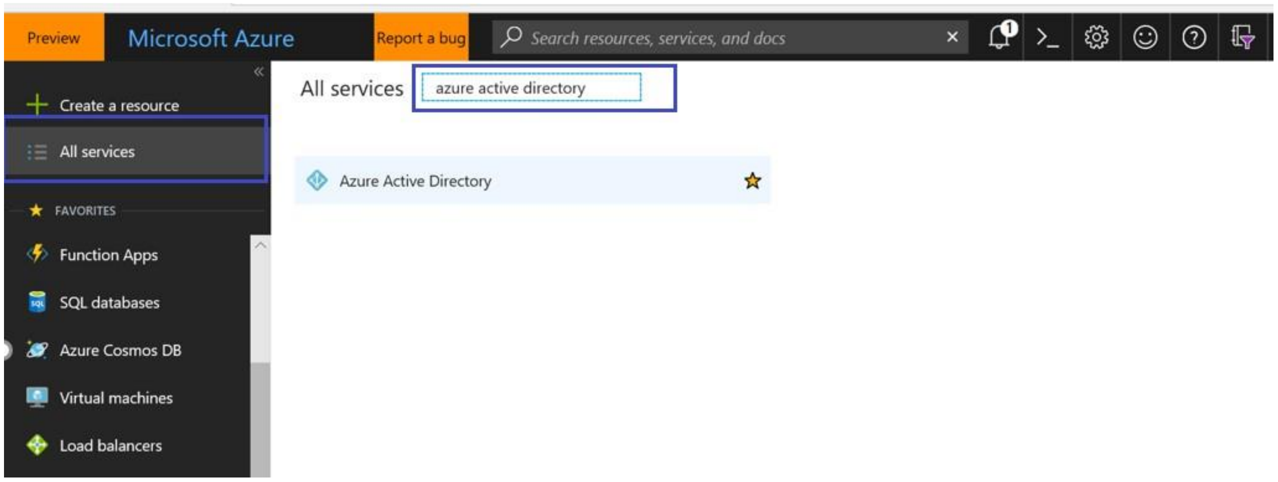
How to setup Azure Active Directory in Azure?

Login to Microsoft Azure portal.

Note: If you do not have an account already. please create a new Azure account. You can start free

Once you are logged into portal, go to all services and search for Azure Active Directory and Select

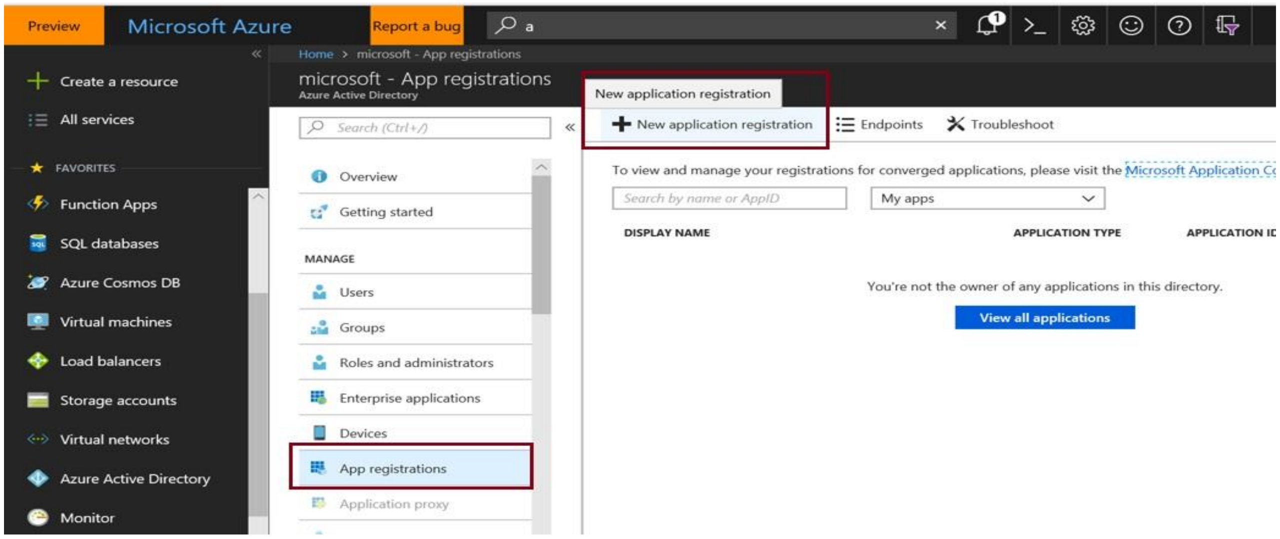
"Azure Active Directory" as shown below.



Create a new tenant for your organization in case it hasn't been created.

https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-access-create-new-tenant

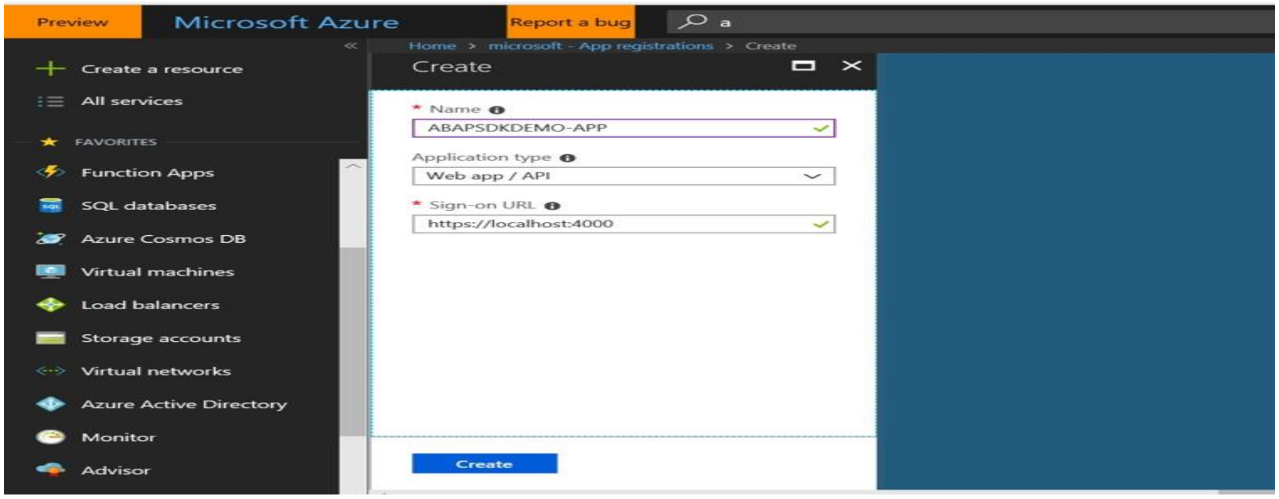
Click on ‘App Registrations’ on left side menu as shown below and click the button ‘New application Registration’



To view and manage your registrations for converged applications, please visit the Microsoft Application C

You're not the owner of any applications in this directory.

Specify details of your Application and press ‘create’ button.

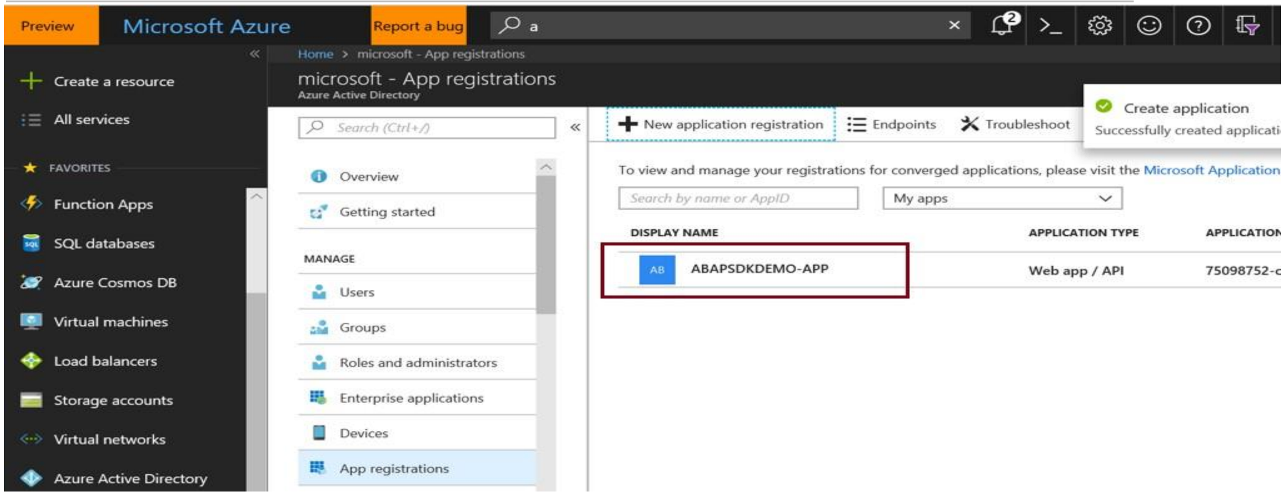


* Name: ABAPSDKDEMO-APP

Application type: Web app / API

* Sign-on URL: https://localhost:4000

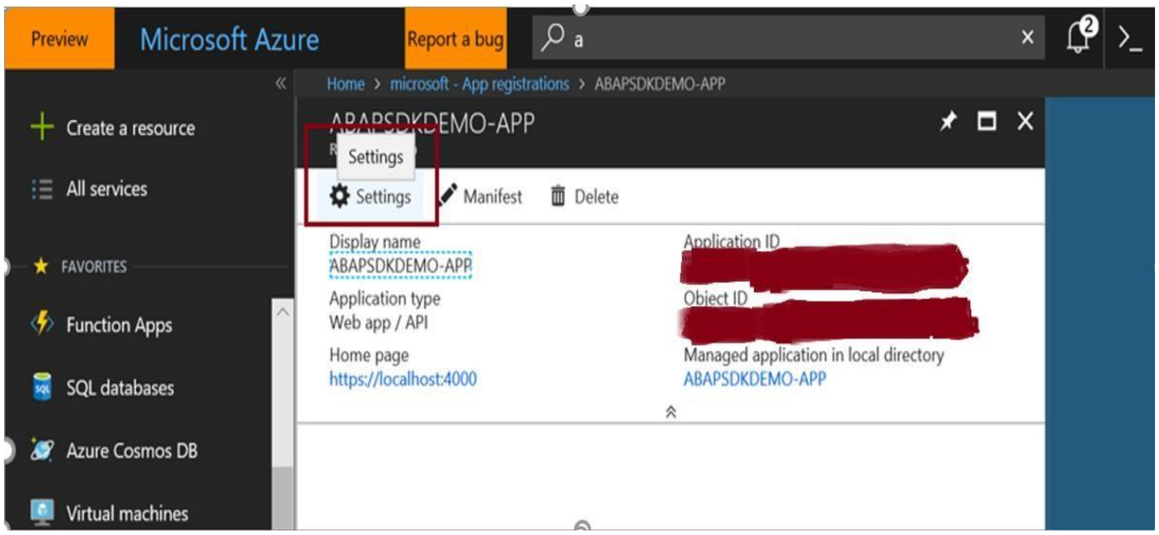
Application is created successfully.



To view and manage your registrations for converged applications, please visit the Microsoft Application

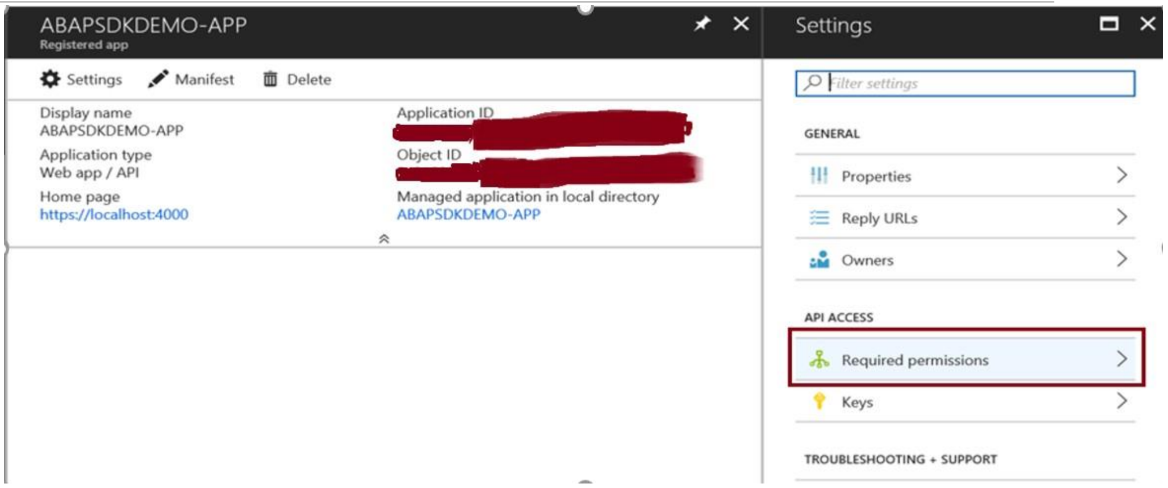
Generate keys for your application

1. Once your application is created, go to your application by clicking on it.
   Copy the application id which will be required in the implementation of code in ABAP SDK. This application id is client Id.



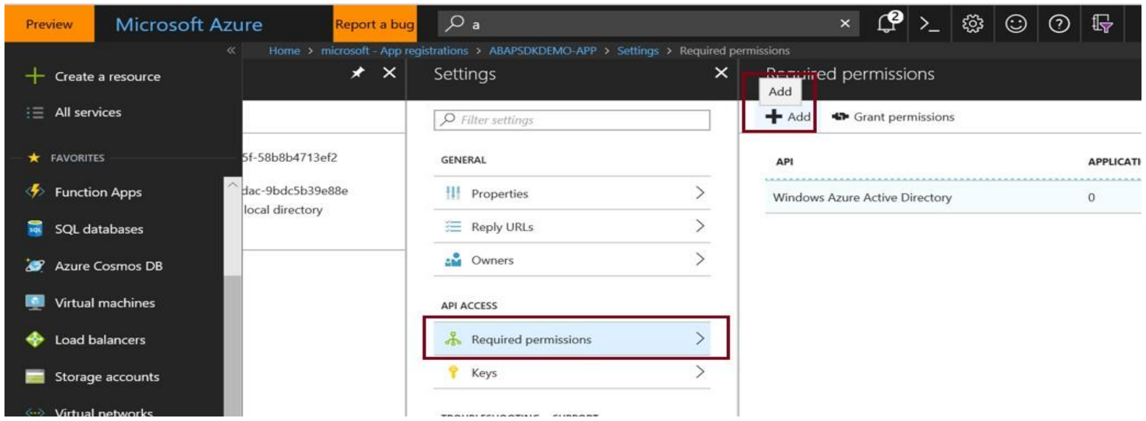
Display name: ABAPSDKDEMO-APP
Application type: Web app / API
Home page: https://localhost:4000
Application ID: [redacted]
Object ID: [redacted]
Managed application in local directory: ABAPSDKDEMO-APP

Click on 'Settings' in the above screen, go to the 'Required Permissions' under API Access as shown below.



Display name
ABAPSDKDEMO-APP
Application type
Web app / API
Home page
https://localhost:4000
Application ID
[redacted]
Object ID
[redacted]
Managed application in local directory
ABAPSDKDEMO-APP

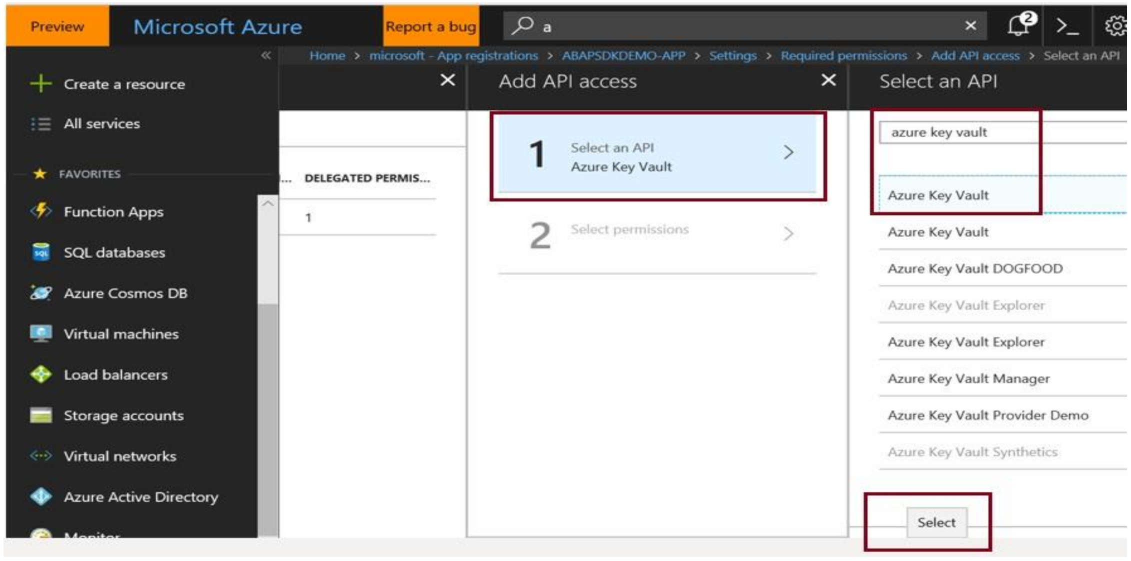
2. Then click on 'Add' button as shown in the below screen.



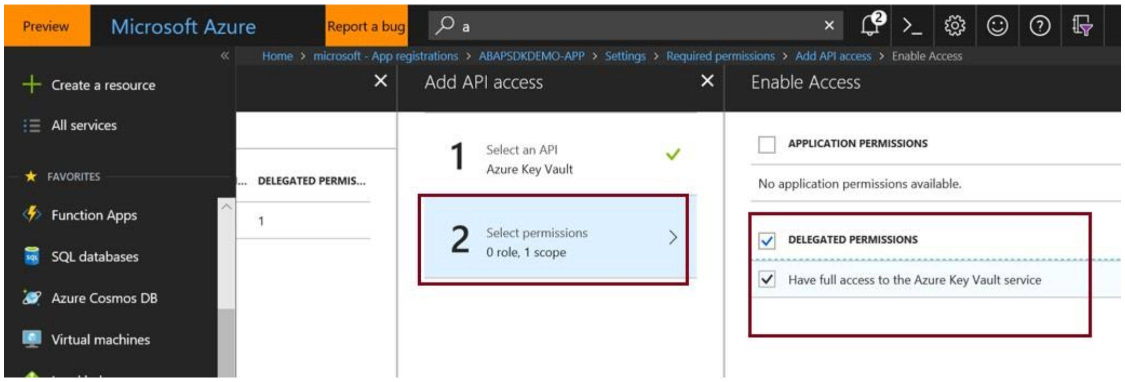
API	APPLICATI
Windows Azure Active Directory	0

3. Under 'Add API access' section click on 'select an API 'and Search Azure Key Vault and Select the same as shown below.

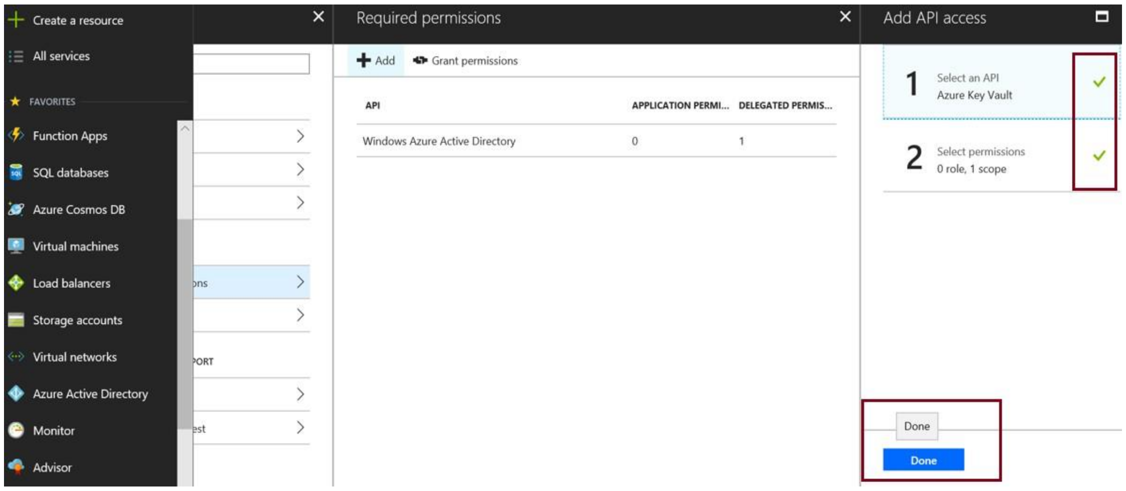
Note:
In this step, we have chosen Azure Key vault as an external application as an example.
In real world scenario, you need to choose your existing API which you want to access with AAD token for authentication.



4. Under Add API access section, click on 'Select permissions' and enabled the checkbox for 'Delegated Permissions' as shown below.

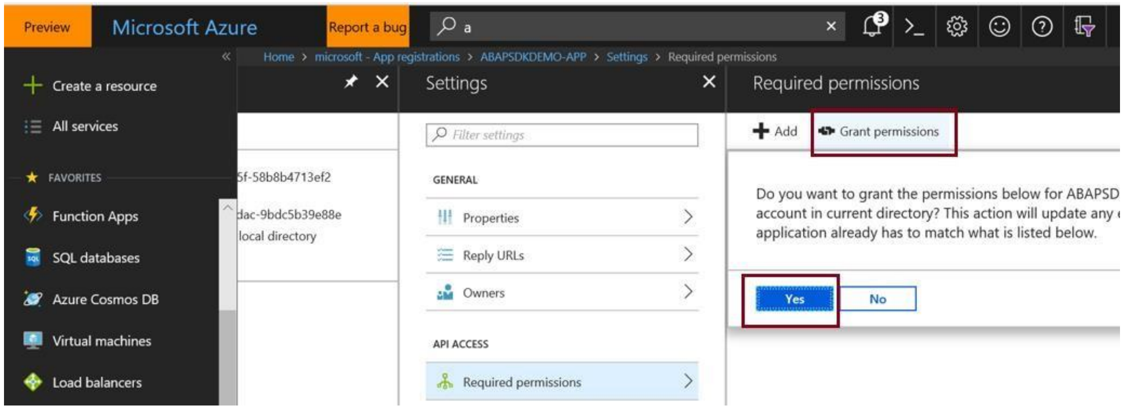

Preview
Microsoft Azure
Report a bug
Home > microsoft - App registrations > ABAPSDKDEMO-APP > Settings > Required permissions > Add API access > Enable Access
Create a resource
All services
FAVORITES
Function Apps
SQL databases
Azure Cosmos DB
Virtual machines
DELEGATED PERMIS...
Add API access
1 Select an API Azure Key Vault
2 Select permissions 0 role, 1 scope
Enable Access
APPLICATION PERMISSIONS
No application permissions available.
DELEGATED PERMISSIONS
Have full access to the Azure Key Vault service

Then click on 'Done' button to complete the Required permissions activity.



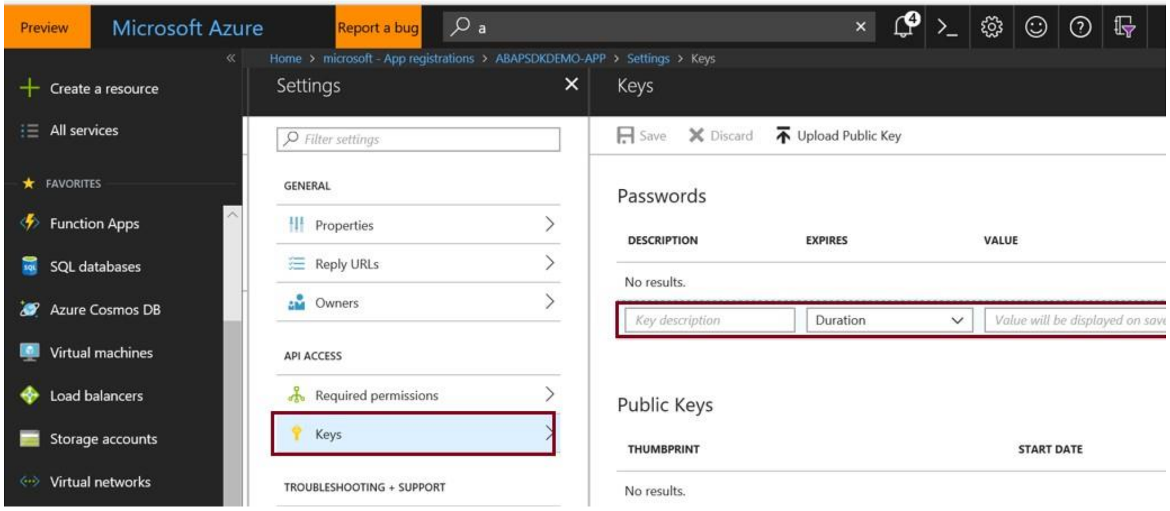
API	APPLICATION PERMI...	DELEGATED PERMIS...
Windows Azure Active Directory	0	1

5. Click on 'Grant permissions' and press 'Yes' button as shown below.



Do you want to grant the permissions below for ABAPSD account in current directory? This action will update any application already has to match what is listed below.

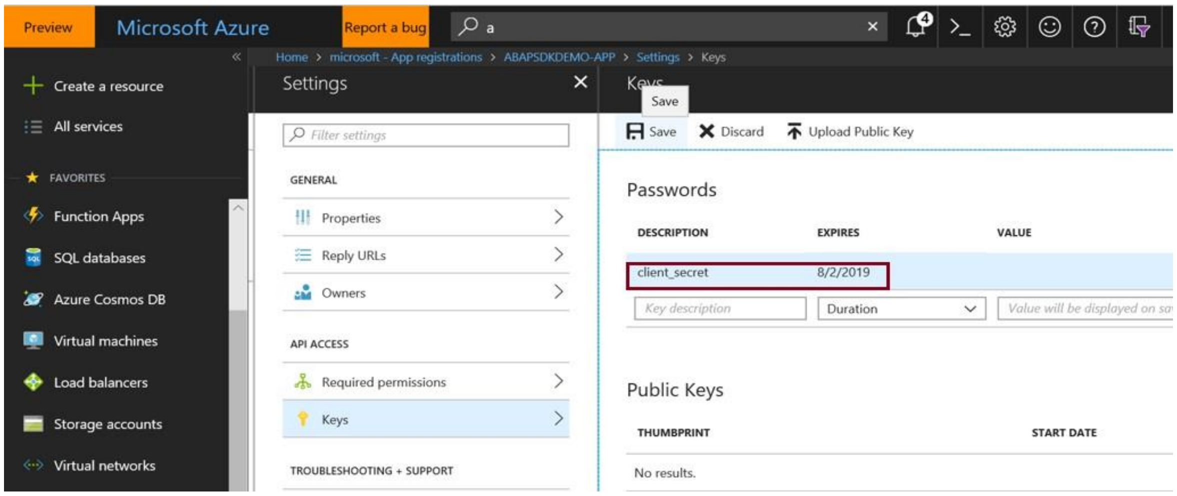
6. Click on 'Keys' section and provide key description and expiry Duration in the below screen. Under 'EXPIRES' dropdown list, you can select the expiry duration.



DESCRIPTION	EXPIRES	VALUE
No results.		

THUMBPRINT	START DATE
No results.	

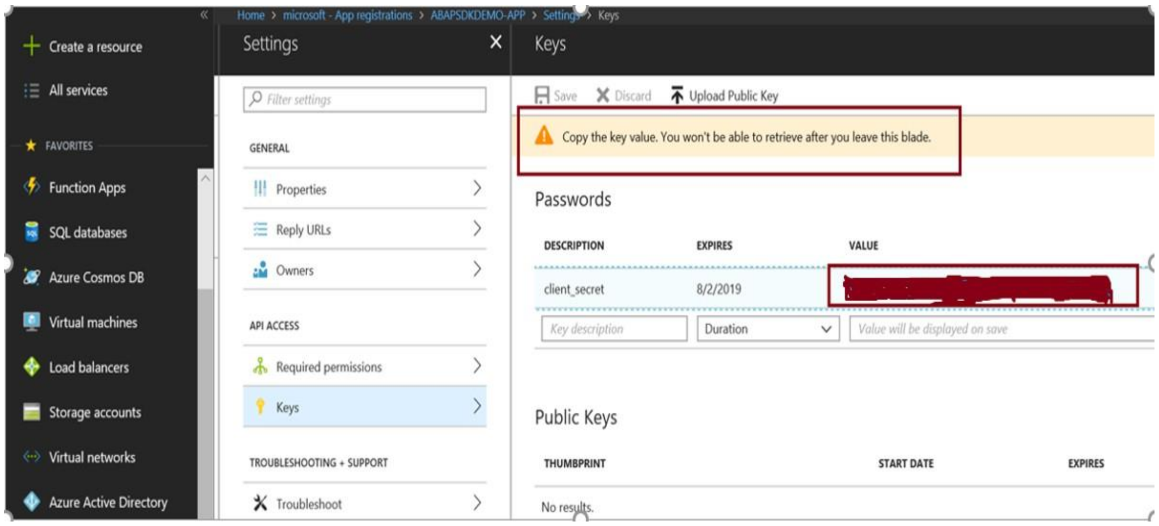
Here key description is provided as client_secret and it expires on 8/2/2019.



DESCRIPTION	EXPIRES	VALUE
client_secret	8/2/2019	

THUMBPRINT	START DATE
No results.	

7. Click on 'Save' button to generate secret key. **Copy this key and it will be used in ABAP SDK implementation. Please remember you won't be able to retrieve this key once you leave the screen.**



Copy the key value. You won't be able to retrieve after you leave this blade.

Passwords

DESCRIPTION	EXPIRES	VALUE
client_secret	8/2/2019	[redacted]
Key description	Duration	Value will be displayed on save

Public Keys

THUMBPRINT	START DATE	EXPIRES
No results.		

Steps to use AAD authentication from SAP using ABAP SDK for Azure

Creation of RFC destination to Azure Active Directory

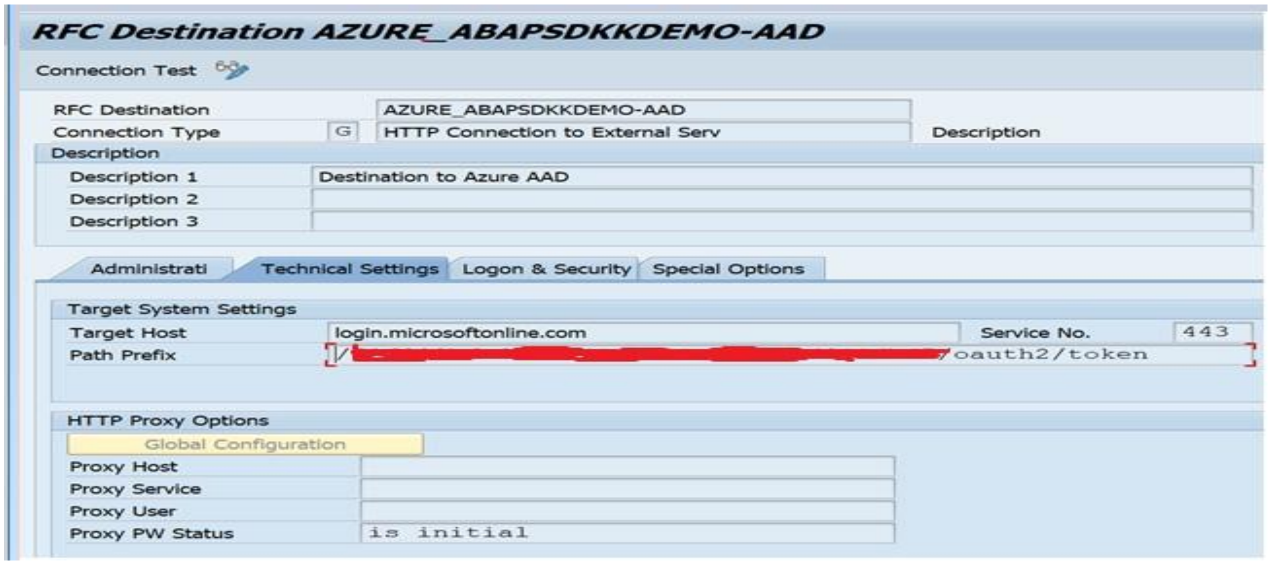
Go to transaction SM59 in your SAP system and create new RFC destination of type ‘G’. Maintain your Azure Active directory endpoint in the Target host and path prefix for authorization token as shown below.

Target host: **login.microsoftonline.com**

Port: **443**

Path Prefix: **/<Input Tenant ID>/oauth2/token**

For Tenant ID details and creating a new tenant id in Azure Active Directory, please refer this document section ‘How to setup Azure Active Directory in Azure?’



RFC Destination AZURE_ABAPSDKKDEMO-AAD

RFC Destination	AZURE_ABAPSDKKDEMO-AAD	
Connection Type	G HTTP Connection to External Serv	Description

Description

Description 1	Destination to Azure AAD
Description 2	
Description 3	

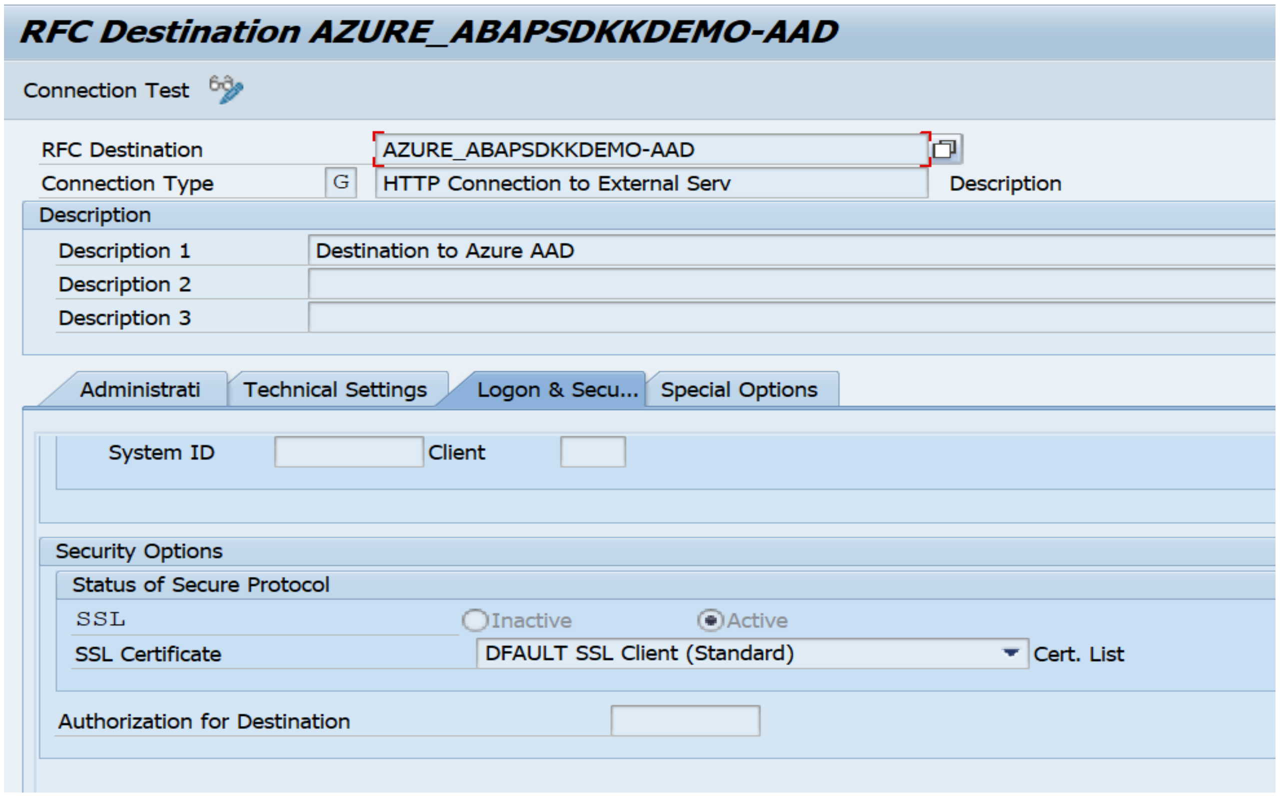
Target System Settings

Target Host	login.microsoftonline.com	Service No.	443
Path Prefix	/[redacted]/oauth2/token		

HTTP Proxy Options

Proxy Host	
Proxy Service	
Proxy User	
Proxy PW Status	is initial

Now go to 'Logon & Security' tab and choose radio button SSL 'Active' and select SSL certificate 'DFAULT SSL Client (Standard)'.



RFC Destination AZURE_ABAPSDKKDEMO-AAD

Connection Test

RFC Destination	AZURE_ABAPSDKKDEMO-AAD	
Connection Type	G HTTP Connection to External Serv	Description

Description

Description 1	Destination to Azure AAD
Description 2	
Description 3	

Administrati | Technical Settings | Logon & Secu... | Special Options

System ID | Client

Security Options

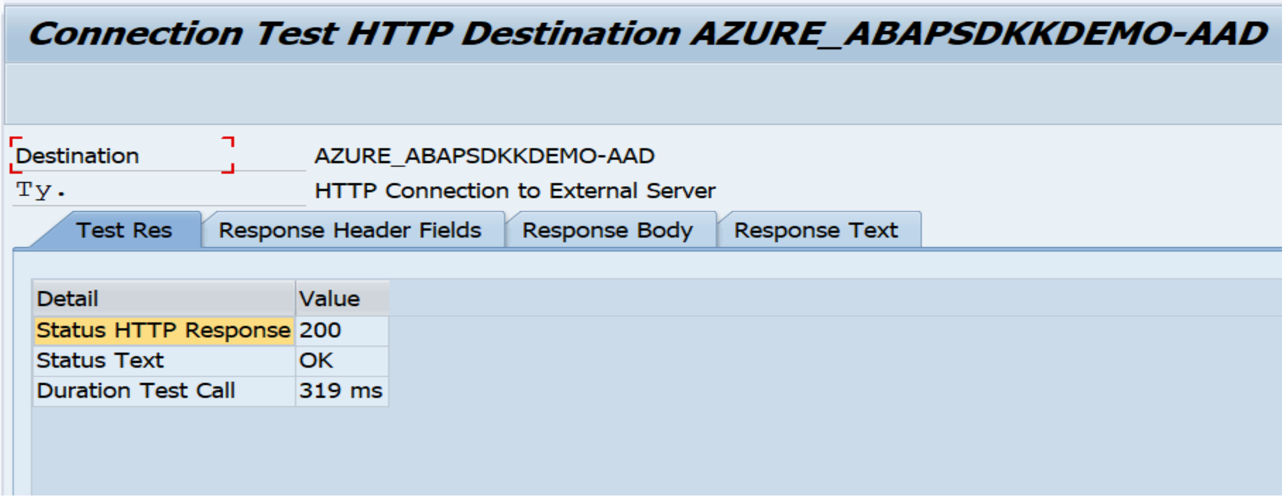
Status of Secure Protocol

SSL ( ) Inactive [✓ Active]

SSL Certificate: DFAULT SSL Client (Standard) Cert. List

Authorization for Destination

Do a connection test to make sure it is working. RFC destination is working.



Connection Test HTTP Destination AZURE_ABAPSDKKDEMO-AAD

Destination	AZURE_ABAPSDKKDEMO-AAD
Ty.	HTTP Connection to External Server

Test Res | Response Header Fields | Response Body | Response Text

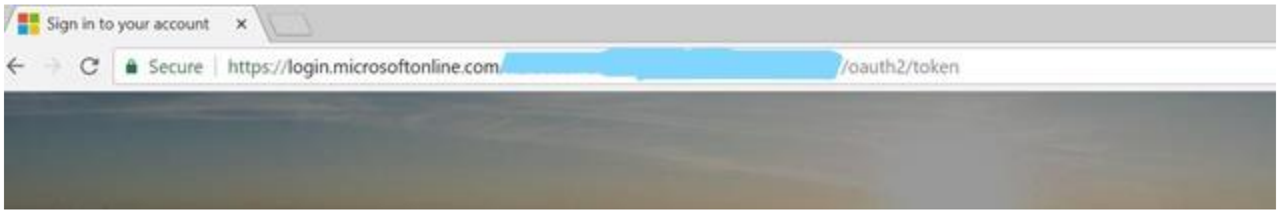
Detail	Value
Status HTTP Response	200
Status Text	OK
Duration Test Call	319 ms

STRUST Setup

We need to import Microsoft's Certificate and import in STRUST for SSL handshake between SAP system and Azure Active Directory over HTTPS protocol. To download the certificate, in your browser, go to URL with the hostname and path prefix you used for creating RFC destination.

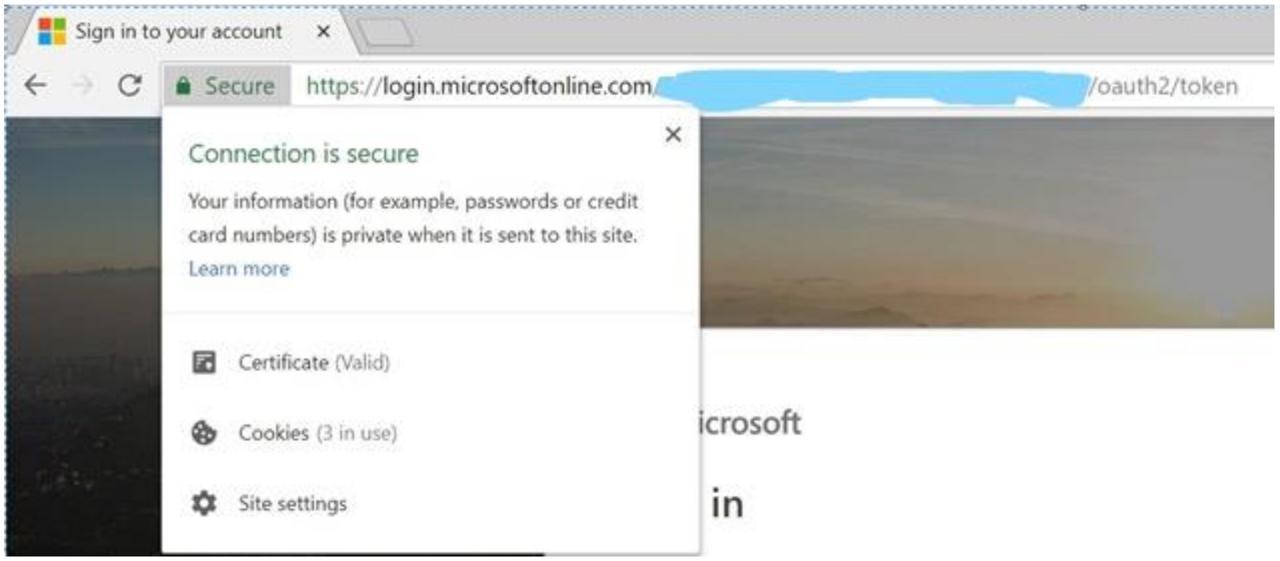
Target host: **login.microsoftonline.com**

Path Prefix: **/<Input Tenant ID>/oauth2/token**



https://login.microsoftonline.com/[redacted]/oauth2/token

Click on the Lock symbol you find next to refresh button in Chrome browser and select Certificate to view the certificate used for communication



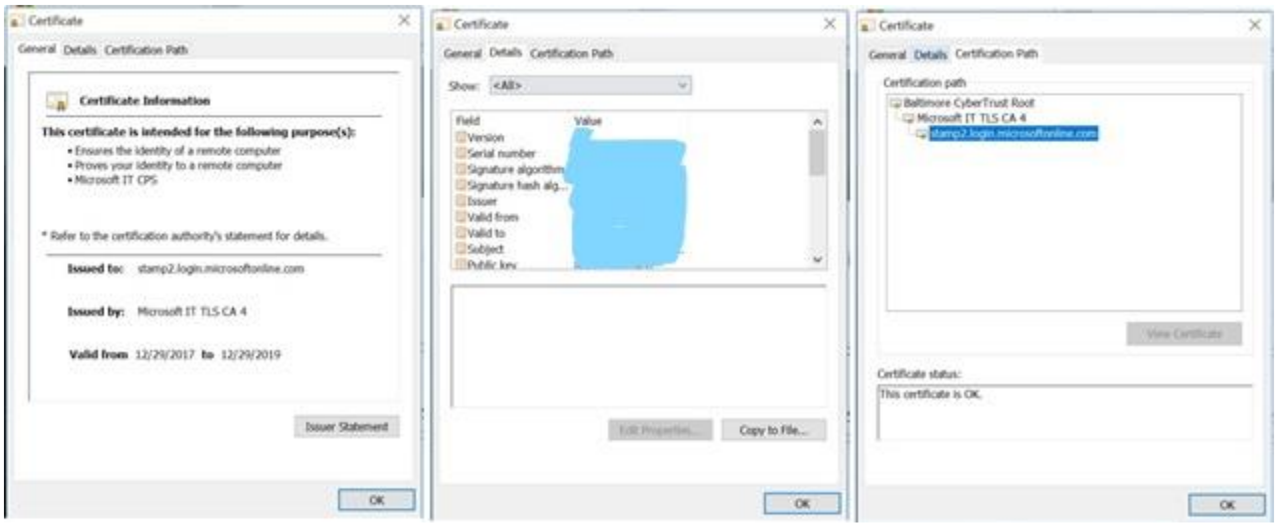
https://login.microsoftonline.com/[redacted]/oauth2/token

Connection is secure

Your information (for example, passwords or credit card numbers) is private when it is sent to this site.

In the certificate, go to Details tab and Choose button 'Copy to File' to download the certificate to your local machine. Repeat the process and download all the certificate until root. In this case, you need to download two certificates.

1. Microsoft IT TLS CA 4
2. Baltimore Cyber Trust Root



Certificate Information

This certificate is intended for the following purpose(s):
- Ensures the identity of a remote computer
- Proves your identity to a remote computer
- Microsoft IT CPS

* Refer to the certification authority's statement for details.

Issued to: stamp2.login.microsoftonline.com

Issued by: Microsoft IT TLS CA 4

Valid from 12/29/2017 to 12/29/2019

[redacted]

Baltimore CyberTrust Root
Microsoft IT TLS CA 4
stamp2.login.microsoftonline.com

This certificate is OK.

when all the certificates are downloaded, Go to STRUST transaction in your SAP system and Import all these certificates in DFAULT PSE.

Note: We are not going through the process of Importing certificates in STRUST in this document. It is straight forward, and your BASIS team can help you to do this activity.

Configuration

ABAP SDK has following main configuration tablesand they need to be maintained. We will create a new Interface ID to establish connection between SAP system and target Azure Active Directory (AAD). A new Interface ID needs to be created for each AAD namespace.

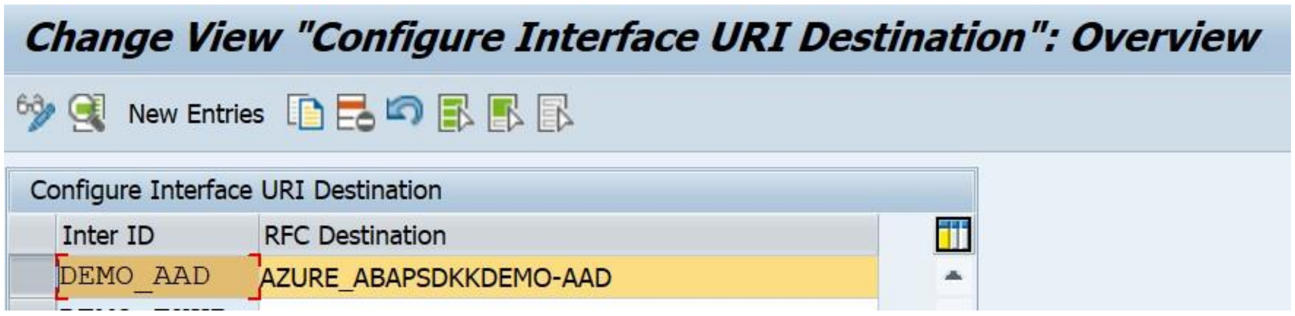
ZREST_CONFIG – Master Table for Interface ID Maintenance. You must define a new Interface name and maintain the RFC destination that was created for target Event hub.

ZREST_CONF_MISC – This is an Interface Miscellaneous table which contains information on Alerts and re-processing of failed messages automatically.

ZADF_CONFIG – This is an Interface extension table. This stores data that is more specific to Azure Services like SAS keys, AAD secrets and processing Method.

ZREST_CONFIG

Create a new Interface ID like 'DEMO_AAD' and Maintain the RFC destination you created earlier.


Change View "Configure Interface URI Destination": Overview

Configure Interface URI Destination

Inter ID	RFC Destination
DEMO_AAD	AZURE_ABAPSDKKDEMO-AAD

ZREST_CONF_MISC

Create an entry in table 'ZREST_CONF_MISC' for the above interface Id 'DEMO_AAD'.

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')

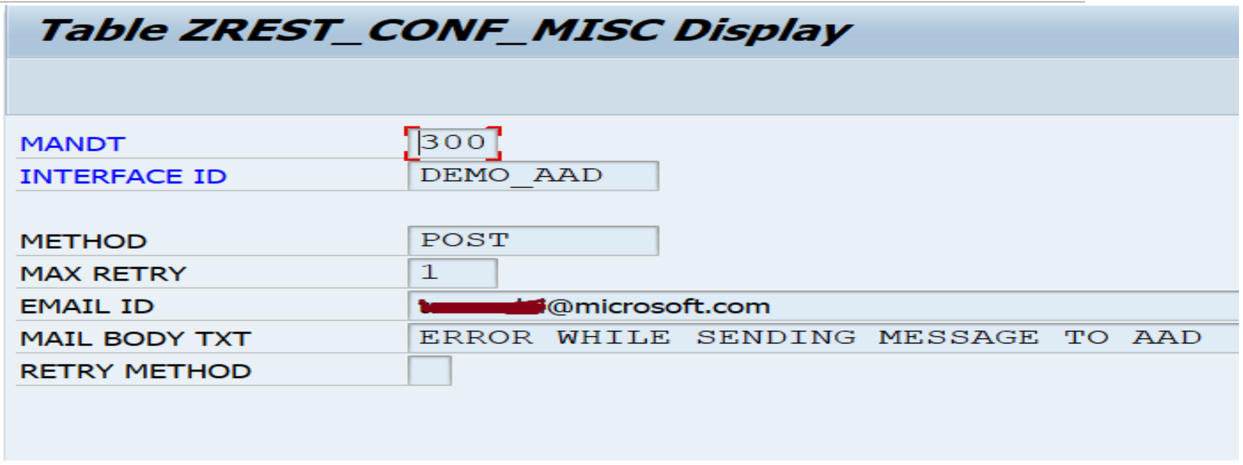


Table ZREST_CONF_MISC Display

MANDT	300
INTERFACE ID	DEMO_AAD
METHOD	POST
MAX RETRY	1
EMAIL ID	[redacted]@microsoft.com
MAIL BODY TXT	ERROR WHILE SENDING MESSAGE TO AAD
RETRY METHOD	

ZADF_CONFIG

Create an entry in table 'ZADF_CONFIG' for the above interface Id 'DEMO_AAD'.

<u>Details of configuration:</u>

- INTERFACE_TYPE is 'Azure Active Directory'.
- SAS_KEY is the shared access key. This is the key which generated in AAD under section 'Generate keys for your application' Step 7 (refer Page 8). You need to change this key in this config table whenever key is changed in Azure.
- URI is left blank. This may be required for future versions.
- SERVICE_TYPE can be synchronous(S) or asynchronous(A)
- IS_TRY is a reprocessing flag, maintain as blank. it can be configured for reprocessing in case of failure of services.

Note: This field can be utilized in our future release to control the reprocessing based on value of X. Presently it's should be enabled as blank.

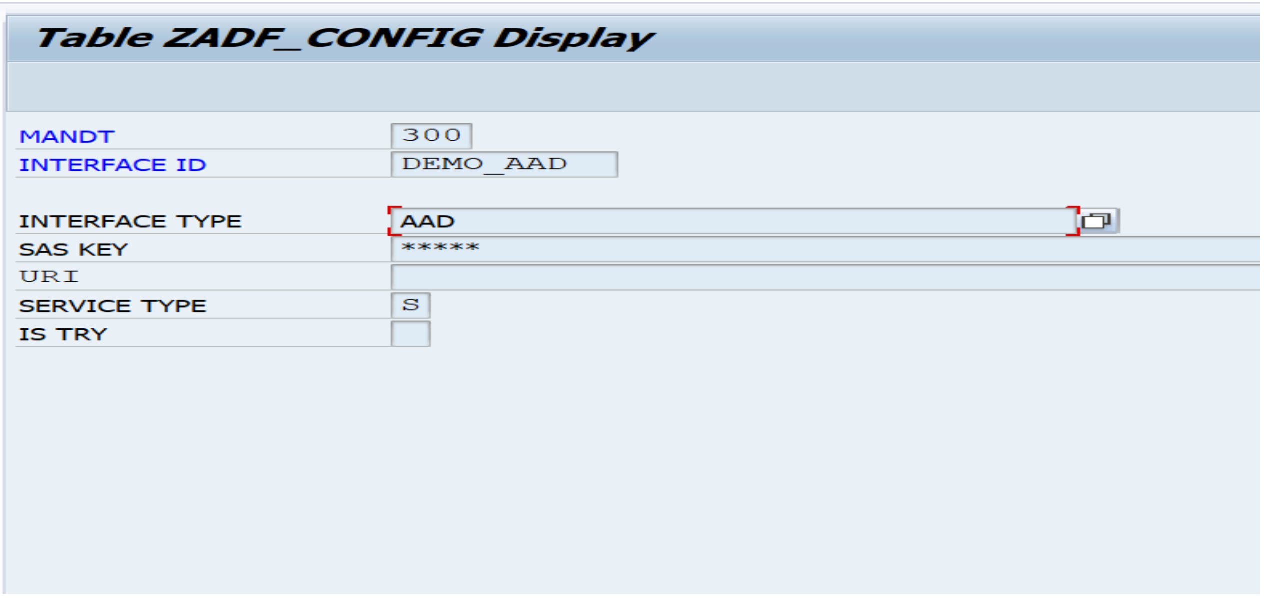


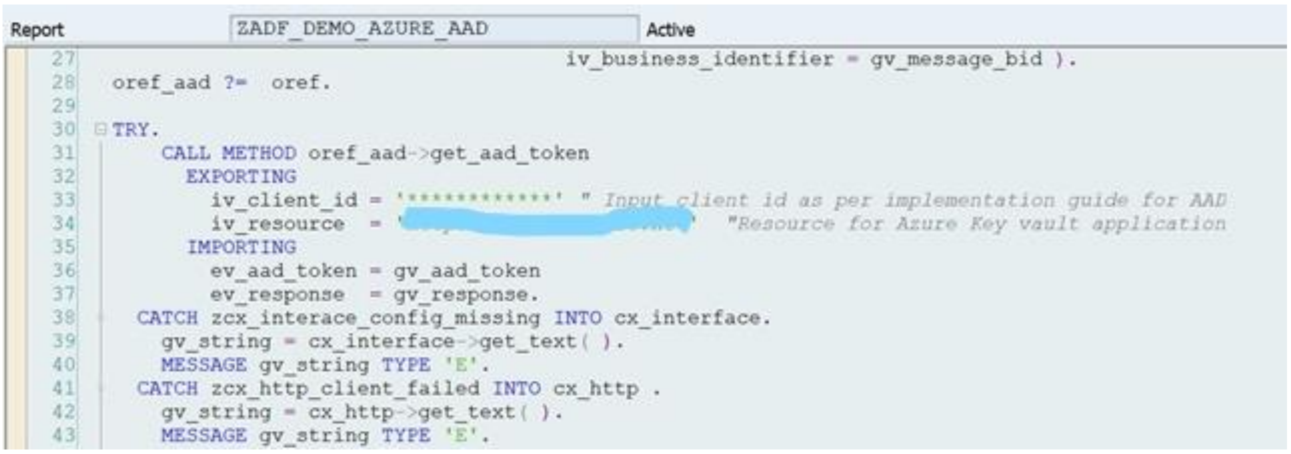
Table ZADF_CONFIG Display

MANDT	300
INTERFACE ID	DEMO_AAD
INTERFACE TYPE	AAD
SAS KEY	*****
URI	
SERVICE TYPE	S
IS TRY	

DEMO Program

Please refer to DEMO program ‘ZADF_DEMO_AZURE_AAD’ to generate AAD token for AAD based authentication.

Please note that in the demo program, application id generated in step 1 of Generate keys for your application is used as client id.

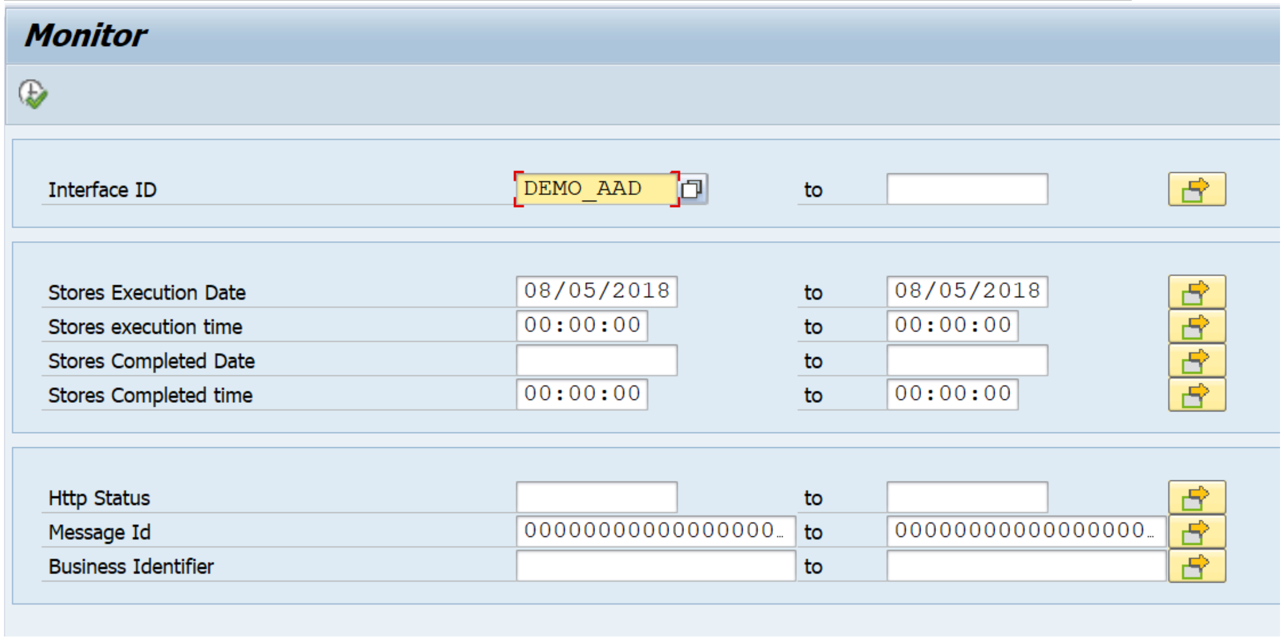


```
Report    ZADF_DEMO_AZURE_AAD    Active

                                       iv_business_identifier = gv_message_bid ).
  oref_aad ?=  oref.

TRY.
    CALL METHOD oref_aad->get_aad_token
      EXPORTING
        iv_client_id = '*************' " Input client id as per implementation guide for AAD
        iv_resource  = '[redacted]'  "Resource for Azure Key vault application
      IMPORTING
        ev_aad_token = gv_aad_token
        ev_response  = gv_response.
  CATCH zcx_interace_config_missing INTO cx_interface.
    gv_string = cx_interface->get_text( ).
    MESSAGE gv_string TYPE 'E'.
  CATCH zcx_http_client_failed INTO cx_http .
    gv_string = cx_http->get_text( ).
    MESSAGE gv_string TYPE 'E'.
```

ABAP SDK Monitor

We have provided an Interface Monitor (Transaction ZREST_UTIL), using this monitor you can view history of all the messages that were posted to Azure Services. In case you have a scheduled a background job to post messages to Azure, you can view the statuses of the messages in this monitor. This Monitor can be used for troubleshooting and re-processing of the message as well.

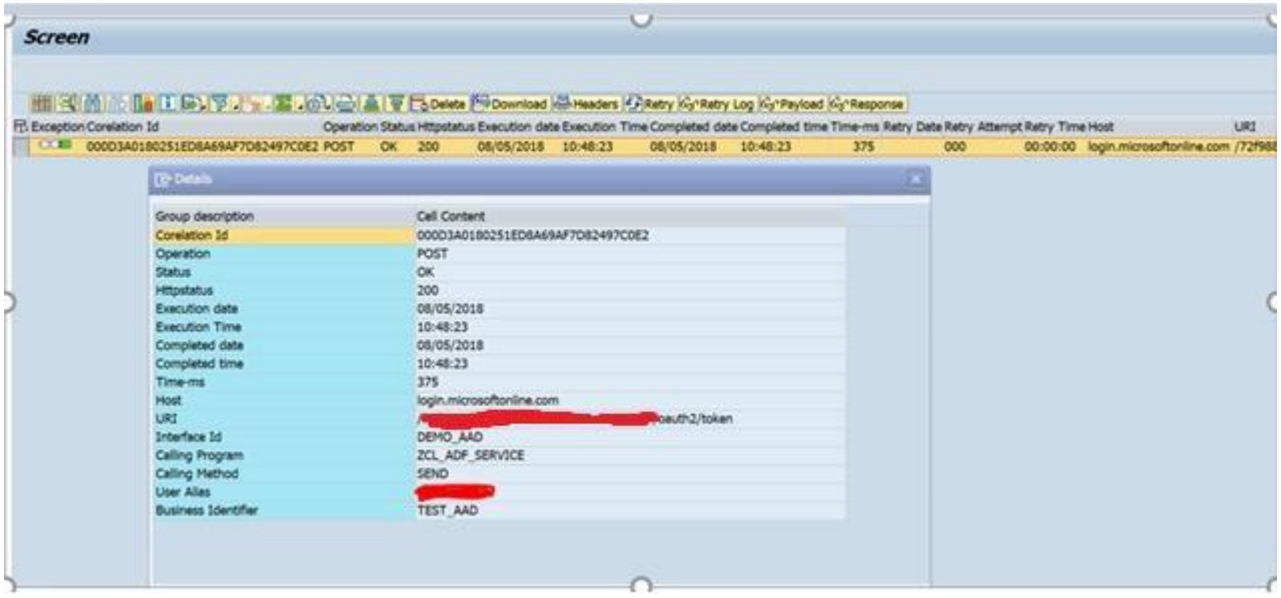
Go to transaction ZREST_UTIL and provide your Interface ID in the selection screen and execute to view all the messages



Monitor

Interface ID	DEMO_AAD	to	
Stores Execution Date	08/05/2018	to	08/05/2018
Stores execution time	00:00:00	to	00:00:00
Stores Completed Date		to	
Stores Completed time	00:00:00	to	00:00:00
Http Status		to	
Message Id	00000000000000000…	to	00000000000000000…
Business Identifier		to	

In this monitor, you can view the status of the HTTPs message and its headers, response, payload and so on. In case of errors, you can also re-process the message from this tool.



Screen

Exception	Corelation Id	Operation	Status	Httpstatus	Execution date	Execution Time	Completed date	Completed time	Time-ms	Retry Date	Retry Attempt	Retry Time	Host	URI
	000D3A0180251ED8A69AF7D82497C0E2	POST	OK	200	08/05/2018	10:48:23	08/05/2018	10:48:23	375		000	00:00:00	login.microsoftonline.com	/72f988

Details

Group description	Cell Content
Corelation Id	000D3A0180251ED8A69AF7D82497C0E2
Operation	POST
Status	OK
Httpstatus	200
Execution date	08/05/2018
Execution Time	10:48:23
Completed date	08/05/2018
Completed time	10:48:23
Time-ms	375
Host	login.microsoftonline.com
URI	/[redacted]oauth2/token
Interface Id	DEMO_AAD
Calling Program	ZCL_ADF_SERVICE
Calling Method	SEND
User Alias	[redacted]
Business Identifier	TEST_AAD

Auto re-processing of failed messages

For auto-processing of messages in case of failures, you must schedule a background job for program 'ZREST_SCHEDULER' as a pre-requisite.