

**Objective** - To Simulate SSO login when a SaaS platform is integrated with Azure Entra ID.

**Status Quo** -

1. We have a SaaS platform, where the users are created in user management and only the user created in the platform can login.
2. In the SaaS platform, users data is stored in Mysql database.
3. A Clients want to integrate the SaaS platform with their own CRM
4. Clients will login to their CRM and once logged in the users will see a link to visit the SaaS platform. Once the user clicks on it, the user login via SSO.


## Overview Terms

Terms	Meaning	Where to configure	Who will provide to whom
SP (Service Provider)	Your app – <a href="#">abc-platform</a>	In azure , this is the identity of our app in azure	SP will create it and provide it to Azure Entra ID > Enterprise Application > Single Sign-On settings
IdP (Identity Provider)	Microsoft Entra ID	This is the unique id created by azure identity of you app within azure	Auto generated, azure to provide. It gets validated on the SP side during the SAML response with the certificate.
ACS URL	Where the IdP sends the SAML response: <a href="https://abc-platform.onrender.com/login/callback">https://abc-platform.onrender.com/login/callback</a>	This is the url where azure will send the SAML response	SP To provide to azure >> application
Entity ID (SP)	Unique ID for your app: <a href="#">urn:abc-platform</a>	This is the identity of our app	SP will create it and configure it in Azure Entra ID under Identifier (Entity ID)
Login URL (IdP SSO URL)	Used to redirect users to Azure for login: <a href="https://login.microsoftonline.com/.../saml2">https://login.microsoftonline.com/.../saml2</a>	When the user clicks on login, they will be redirected to the Azure login page."	Azure to provide to SP

Login URL (IdP SSO URL)	Auto-config URL with IdP SAML settings (certificates, endpoints, etc.)		Azure generated
SAML Certificate	Used to sign and validate assertions	SAML certificate generated in azure	Azure generates the certificate. The SP (your app) must configure this Base64 certificate in the SAML configuration (e.g., in <code>server.js</code> or wherever your app validates SAML responses).

Step 1: User opens your login page


URL used:

 <https://abc-platform.onrender.com/login>

What happens:

This is the **SP-Initiated** login. Your frontend or backend redirects the user to the IdP login URL.

## Step 2: Redirect to Azure AD (IdP)


- Redirected to IdP Login URL:  
 <https://login.microsoftonline.com/a8757937-5c70-468d-a17c-f6a374cc7803/saml2>
- What happens:
  - A SAML authentication request is created by your app.
  - It contains the SP Entity ID (urn:abc-platform), the ACS URL (/login/callback), and a request for user attributes.
  - This request is sent to the **Microsoft Entra ID Login URL**.

## Step 3: User authenticates in Azure

- What happens:

- Users log into Microsoft Entra ID with their enterprise credentials.
- After successful authentication, Azure builds a **SAML Response**.
- The SAML Response:
  - Contains user attributes like:
    - givenname = user.givenname
    - surname = user.surname
    - emailaddress = user.mail
  - It is **signed using the SAML certificate** issued in the metadata file.

#### Step 4: SAML Response sent to your ACS URL

- URL used:
  -  <https://abc-platform.onrender.com/login/callback> (ACS – Assertion Consumer Service)
- What happens:
  - Azure redirects the browser back to this URL with the SAML Response (usually POSTed).
  - Your backend **parses and validates** the SAML Response using:
    - The certificate from Azure (fetched via metadata or configured manually).
    - The Microsoft Entra Identifier (IdP entity ID):  
<https://sts.windows.net/a8757937-5c70-468d-a17c-f6a374cc7803/>
- Backend Actions:
  - Validate the signature of the assertion.
  - Extract attributes (name, email, etc.).
  - Create a session or JWT token for the user.
  - Redirect the user to the dashboard.

## Step 5: User is logged in and redirected

- What happens:
  - Your app now considers the user authenticated.
  - A session or secure token is set.
  - The user is redirected to a dashboard or home page.

## Optional: Logout Flow

- Logout URL:
- <https://login.microsoftonline.com/a8757937-5c70-468d-a17c-f6a374cc7803/saml2>
- If your app supports **Single Logout (SLO)**, it can:
  - Send a logout request to this URL.
  - Azure logs the user out of all sessions.

## Metadata URL

- <https://login.microsoftonline.com/.../federationmetadata.xml?...>
- Purpose:
  - This file contains all IdP configuration needed by your app:
    - IdP SSO URL
    - Certificate (Base64)
    - IdP Entity ID
    - Binding protocols
- Your app (SP) should:
  - Either fetch this file dynamically
  - manually configure these values in your SAML library

**The Question** is How the SaaS platform will identify the Azure user is the same user whose credentials are created in the SaaS platform?

**Answer**

So, the answer is when the SAML assertion is sent by azure to SP (IdP to SP) attribute like email id will be sent. SP will decode the SAML assertion and find the users email id.

Now SP will be checked in the Mysql database if there is a user available with the same email id , if yes then user will be able to login successfully, else “ there will be a page sorry “you are not authorized, kindly contact admin” message will appear on the screen.