# Authentication integration with organisation identity provider



## Intro

This manual page is mainly for area owners, key customer contacts, but also for end-users so that they can be aware of such solution and can ask management or proper ICT contacts to implement it also in their environment.

TDS platform does not need any external users source as it has it own identity with users database. Every TDS user has account in TDS with his own password and can enable MFA/2FA for better security.

However if organisations would like to make life easier for users by allowing them to be singned-in/authenticated seamlessly just by clicking "OrganisationX AzureAD" button instead of entering credentials, they can decide to integrate TDS platform into AzureAD or similar kind of identity provider.

## Azure AD integration with TDS Keycloak using OpenID Connect

OpenID Connect (OIDC) is quite straightforward and modern solution for adding support for centralised authentication. We usually meet together with customer organisation representatives and their identity provider administrators and settle down procedure details and requirements. Then AzureAD adminstrators need to follow steps how to integrate TDS Keycloak using OIDC as defined in this chapter.

### Registering application in Azure portal

Azure portal is expected to be used: https://portal.azure.com/#home

- Go to https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps
- Click "New registration"
- Set name "Tietoevry DevOps Space (TDS) Keycloak" or something similar
- Select who can use it, usually first option "Accounts in this organizational directory only (for example Tietoevry only - Single tenant)" is OK
- Set Redirect URI to callback URL:

  ```
  https://identity.core.tds.CUSTOMERX.com/auth/realms/tds/broker/main-oidc/endpoint
  ```

  ⚠ It says that it is optional, but it is mandatory in our case.

- Click "Register"
- On displayed "Overview" screen you shall find and copy following values that will be needed later for setup on Keycloak side:
  - Application (client) ID
  - Directory (tenant) ID
- Following claims need to enabled:
  - sAMAccountName (onpremisessamaccountname)
  - email
  - firstname
  - lastname
- Generate Client Secret
  - Go to "Certificates & secrets"
  - Click "New client secret"
    - Provide some description like "keycloak" or "CustomerX Keycloak"
    - Make secret valid for as long period as applicable or possible - usually maximum number of months that organisation allows or even infinity.
      - Remember to generate new secrets on regular basis as security measure. Implementing process for replacing secret on regular basis according to validity possibilities is highly recommended.
    - Click "Add"
    - Now you MUST copy "Value" of new secret otherwise it will disappear soon. It will be needed later for setup on Keycloak side.
- AzureAD team must run following powershell commands to properly enable "sAMAccountName" claim
  - First we add new policy for TDS Keycloak in Azure AD, for example "Tietoevry-TDS-Keycloak-Policy" in this case:

```
New-AzureADPolicy -Definition @('{"ClaimsMappingPolicy":{"Version":1,"IncludeBasicClaimSet":"
true", "ClaimsSchema": [{"Source":"user","ID":"givenname","JwtClaimType":"given_name"},{"Source":"
user","ID":"surname","JwtClaimType":"family_name"},{"Source":"user","ID":"mail","JwtClaimType":"
mail"},{"Source":"user","ID":"onpremisessamaccountname","JwtClaimType":"samaccountname"}]}}') -
DisplayName "Tietoevry-TDS-Keycloak-Policy" -Type "ClaimsMappingPolicy"
```

https://docs.microsoft.com/en-us/powershell/module/azuread/new-azureadpolicy?view=azureadps-2.0-preview

- Obtain relevant Enterprise application ID
  - Go to Enterprise Applications https://portal.azure.com/#blade/Microsoft_AAD_IAM/StartboardApplicationsMenuBlade/AppAppsPreview/menuId/
  - Search for application with name that you registered in previous step
- Then we link service principal policy to Enterprise application (always remember to use correct ID):

```
Add-AzureADServicePrincipalPolicy -Id "<Enterprise-Application-UUID>" -RefObjectId "9cb83e39-8b0b-
40da-8e2c-e49b8d2522b7" #'Tietoevry DevOps Space (TDS) Keycloak' & 'Tietoevry-TDS-Keycloak-Policy'
```

https://docs.microsoft.com/en-us/powershell/module/azuread/add-azureadserviceprincipalpolicy?view=azureadps-2.0-preview

- Then go to application permissions and grant "Admin" consent (on behalf of all users in this tenant) as following:

| Microsoft Graph | Claim value | Permission | Type | Granted through | Granted by |
|---|---|---|---|---|---|
| Microsoft Graph | User.Read | Sign in and read user profile | Delegated | Admin consent | An administrator |

- Then go to "Manifest" and enable custom claims mapping as suggested in https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-claims-mapping#security-considerations
  - Find line with "acceptMappedClaims" and change it from "null" to "true"
  - Click "Save"

### Providing information to TDS team

TDS team needs following details in order to finish integration:

| Attribute | Example/template value |
|---|---|
| Authorization URL | https://login.microsoftonline.com/<TENANT-ID>/oauth2/v2.0/authorize |
| Token URL | https://login.microsoftonline.com/<TENANT-ID>/oauth2/v2.0/token |
| Client ID | b1a10f18-5456-47c3-9843-d90cd58c5278 |
| Client Secret | m4yg87DR-_juvp~Tv4U9w-q9x8Mzmo~1Lp |

TDS team then takes over and finishes setup according to internal documentation Identity broker Azure AD OpenID Connect setup#Keycloakpart.

## Google integration with TDS Keycloak using SAML

- Follow official instructions to set up custom SAML application - https://support.google.com/a/answer/6087519?hl=en
  - Metadata will be provided by TDS team, but usually it is at Keycloak URL like https://identity.core.tds.CUSTOMERX.com/auth/realms/tds/broker/google-saml/endpoint/descriptor
- Make sure to set up following attribute mapping:
  - E-mail >> email
  - Windows Login >> sAMAccountName
  - First Name >> FirstName
  - Last Name >> LastName
- Define custom initial page to portal URL, like https://tds.CUSTOMERX.com
- Provide GoogleIDPMetadata.xml metadata file to TDS team.

## Azure AD integration with TDS Keycloak using SAML

Details need to be discussed on meeting with customer representatives and AzureAD administrators.

TDS team recommends OIDC option as its setup and maintenance is generally easier and more flexible than SAML. For example SAML requires proper timing of certificates exchanges when multiple teams/parties must organise themselves in corporate environments - this makes even regular changes more complicated. OIDC requires regular exchanging of secrets, however thanks to possibility to have multiple secrets at the very same time, changes are easier and do not require strict timing like in SAML case when certificates are being exchanged. SAML setup is less straightforward than in OIDC case.

# ADFS integration with TDS Keycloak using SAML

This solution is getting less and less popular in favour of AzureAD, but it is still possible to integrate using ADFS if there is no other way.