Televic Education

ADFS

05 February 2021

Created by Sylvia Joos

televic education

Table of contents

1 Introduction

Active Directory Federation Services (ADFS) is a feature of the Windows Server operating system (OS) that extends end users' single sign-on (SSO) access to applications and systems outside the corporate firewall.

Through SSO capabilities, ADFS can authenticate a user to different, related web apps during a single online session. ADFS shares the user's identity and access rights, also known as claims, across the organization's security boundaries. When users attempt to access a certain web app from one of their trusted business partners -- also known as a federation -- their organization must authenticate the employee's identity information via claims to the host of the web app. The host can then make authorization decisions based on the claims.

With respect to setting up the ADFS coupling, the customer takes the role of the Identity Provider (IdP) and assessmentQ takes the role of the Service Provider (SP).

assessmentQ uses ADFS through Identity Server to authenticate users.

2 Requirements

The requirements are straightforward:

- Support for AD FS 2.0 or higher
- OAuth SHA-2 signature encoding
- Public access to the Identity Provider
- Support for the following claims (will be detailed in the next section)
    - Name Id or Sub
    - E-mail Address
    - Surname
    - Given Name
    - User name *[optional]*
    - Date of Birth *[optional]*
    - UserLanguage *[optional]*

3 Configuration of the Identity Provider

In this section, we will provide sample instructions on how to configure the Identity Provider.

3.1 Configuring claims

In its simplest form, claims are simply statements (for example, name, identity, group), made about users, that are used primarily for authorizing access to claims-based applications located anywhere on the Internet. Each statement corresponds to a value that is stored in the claim.

Claims need to be configured in order to map users between ADFS and assessmentQ.

> ⚠ Important: the user's Windows name needs to be passed without domain prefix as Name ID. In order to accomplish this, the Active Directory Claims Provider Trust need to be changed accordingly.

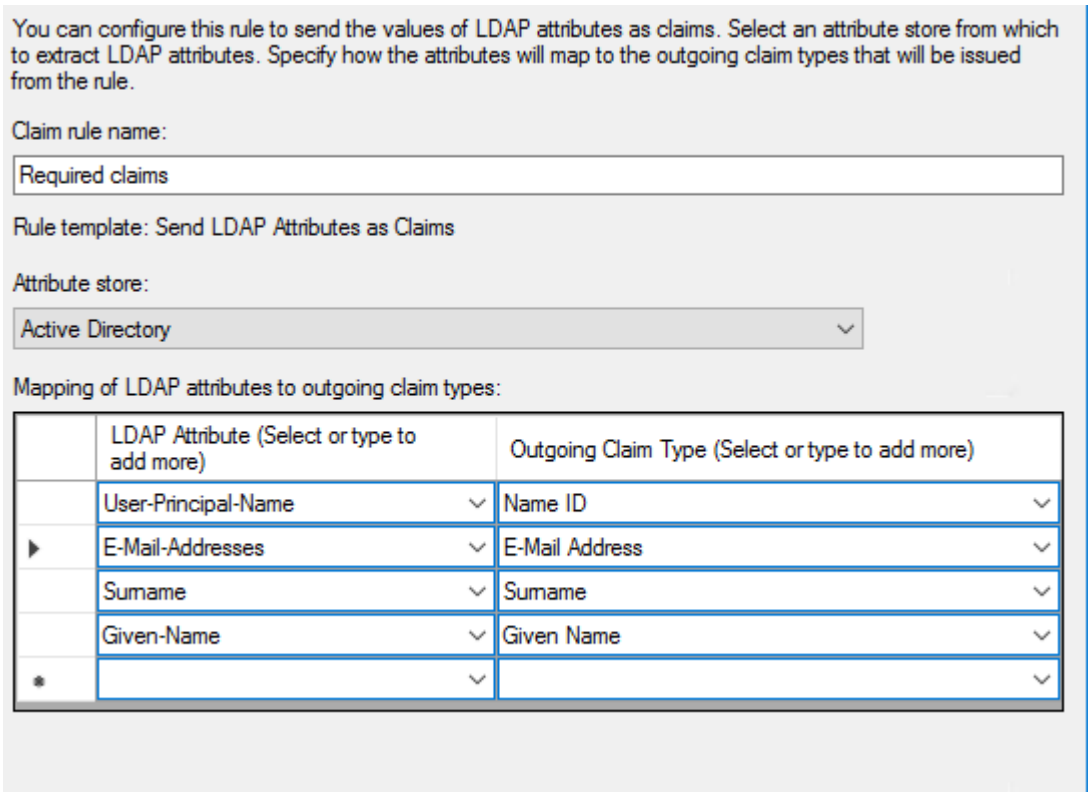
For more information on the role of claims, please visit https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/technical-reference/the-role-of-claims [last visited 01 Mar 2020].

The table below lists the **required** claims.

Name	Description	URI
Name Identifier or Sub	The SAML name identifier of the user	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier
E-Mail Address	The e-mail address of the user	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress
Surname	The surname of the user	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname
Given name	The given name of the user	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname

> ⓘ Name ID Claim can be mapped from anything you wish, as long as it is unique. For example, it is possible to map the SamAccountName to the NameId claim.

Example configuration for mapping the LDAP attributes to the required claims:



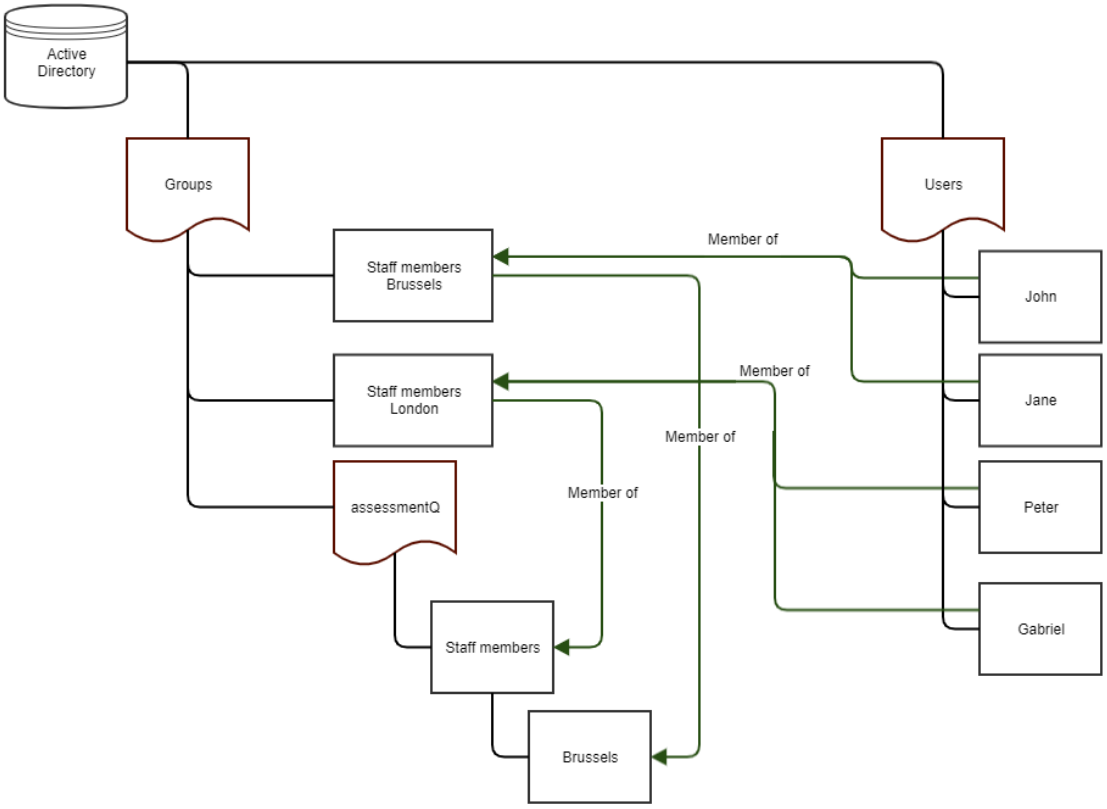
The table below lists the **optional** claims.

Name	Description	URI
User name	The user name of the user	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/name
Date of Birth	The date of birth of the user	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/dateofbirth
UserLanguage	The language of the user *(This claim is defined by Televic)*	http://schemas.televic.com/2017/02/claims/UserLanguage

3.2 Adding support for synchronizing groups between ADFS and assessmentQ

3.2.1 Organizing Groups within the AD

The diagram below depicts the coupling between users and groups in Active Directory (AD) and assessmentQ.



Within the AD structure, you create an Organization Unit (OU) *assessmentQ*. Within this OU, you define all the groups that need to be synchronized with assessmentQ. Subsequently you add the users to these groups, either by linking the user directly to an OU group or by linking an existing AD group to an OU group.

For example:

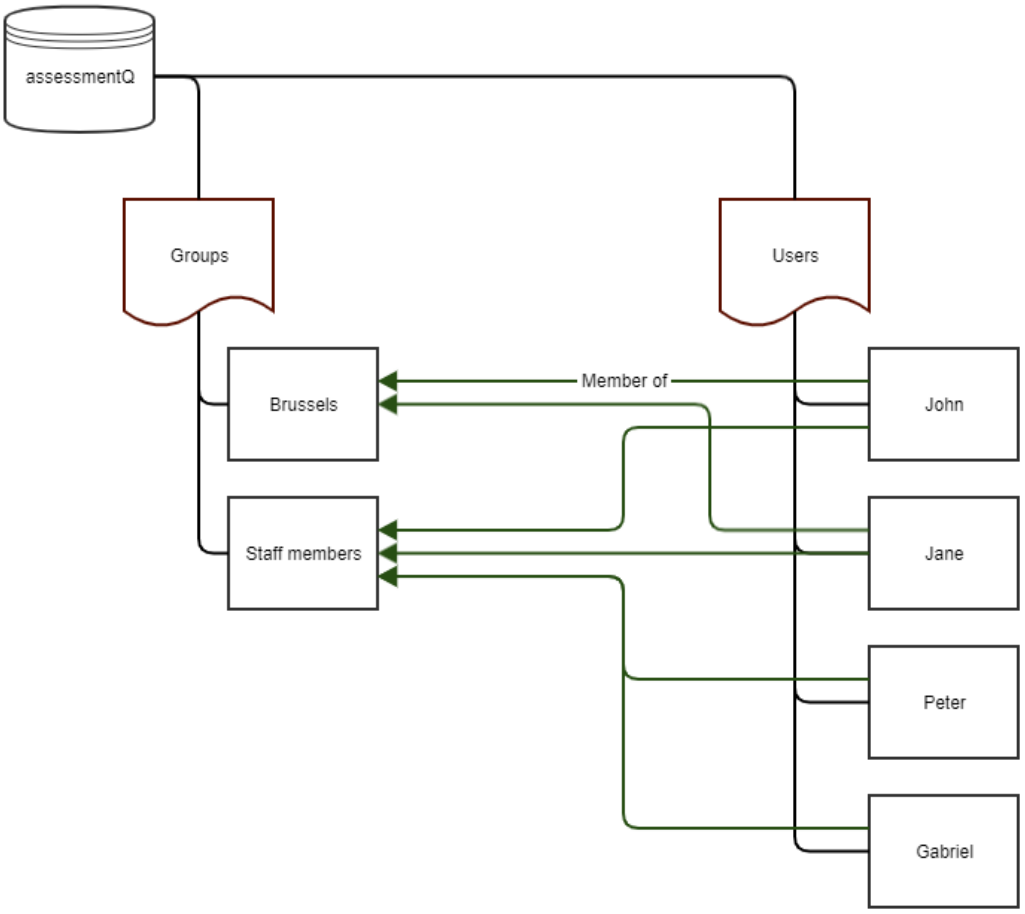
1. John and Jane are members of the AD group *Staff members Brussels*. This group, in turn, is member of the OU group *Brussels*. As a result, John and Jane are members of the OU groups *Brussels* and Staff members within the OU *assessmentQ.*
2. Peter and Gabriel are members of the AD group *Staff members London*. This group, in turn, is member of the OU group *Staff members*. As a result, Peter and Gabriel are members of the OU group *Staff members* within the OU *assessmentQ.*

When John signs in to assessmentQ (via the ADFS coupling), the groups *Staff members* and *Brussels* are automatically created in assessmentQ and John is added to these groups. These groups will not be nested in assessmentQ and will exist next to each other.
When Jane signs in to assessmentQ, she will be added to the groups *Staff members* and *Brussels* (which are already present in the system).

When Peter signs in to assessmentQ, he will only be added to the group *Staff members*. No group *Staff members London* will be created.

As such, groups not part of the OU assessmentQ will not be created in assessmentQ.

In assessmentQ, this will result in the following structure of groups and users:



3.2.2 Limitations

There are some restrictions and limitations related to syncing groups between AD and assessmentQ:

1. Groups within the OU assessmentQ should all have unique names as groups are not nested in assessmentQ.
2. In case groups already exist within assessmentQ, the same name must be used within the OU assessmentQ.
3. If a group name is changed in assessmentQ, it should also be manually changed in the OU assessmentQ and vice versa.
4. Users are never deleted from groups in assessmentQ.
   Example:
   a. In AD, *Jane* is initially member of the group *Staff members.*
   b. *Jane* signs in to assessmentQ (using ADFS). As a result, *Jane* is also added to the group *Staff members* in assessmentQ.
   c. Some time later, in AD, *Jane* is moved to group *Brussels*. Hence, in AD, *Jane* is no longer member of the group *Staff members.*
   d. *Jane* signs in to assessmentQ again (using ADFS). As a result, *Jane* will be added to the group *Brussels*. **Important:** *Jane* is still a member of the *Staff members* group in assessmentQ as well.

   Automatically deleting a user from groups is not supported through ADFS. Also, this could have an impact on reporting and obtained results. As such, if users change groups in AD, they should **manually be removed** from the corresponding assessmentQ group (if needed).

Tip: create a dummy user which is member of all the OU groups and sign in with that particular user in assessmentQ via ADFS. As a result, all assessmentQ groups will be created at once.

3.2.3 Passing on groups to assessmentQ

As detailed in the previous section, the groups passed on the assessmentQ need to be defined in an OU.

To find all groups for a user, which are member of this assessmentQ OU, a custom Attribute Store and 2 custom rules need to be added to the ADFS configuration.

When creating the Attribute store, the type should be set to 'LDAP' and the connection string should point to the full path to the assessmentQ OU (e.g. LDAP://televic-education.com/OU=assessmentQ,DC=televic-education,DC=com).

Next, following two rules should be added.

Claim rule template	Name	Custom rule
Send Claims Using a Custom Rule	Fetch User's Distinguished Name	c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"] => add(store = "Active Directory", types = ("http://schemas.televic.com/2017/02/claims/UserDN"), query = ";distinguishedName;{0}", param = c.Value);
Send Claims Using a Custom Rule	Fetch the User's Groups	c:[Type == "http://schemas.microsoft.com/ws/2008/06/identity/claims/windowsaccountname", Issuer == "AD AUTHORITY"] && c2:[Type == "http://schemas.televic.com/2017/02/claims/UserDN"] => issue(store = "assessmentQ", types = ("http://schemas.televic.com/2017/02/claims/Group"), query = "(&(objectClass=group)(objectCategory=group)(member:1.2.840.113556.1.4.1941:={1}));cn;{0}", param = c.Value, param = c2.Value);

4 Connection details and settings

Important note: The information provided in this section is strictly personal and bound to your assessmentQ environment/channel. Please do not share this information.

The information is this section will allow you to setup an SSO in the assessmentQ **DEMO| PRODUCTION** environment for the channel **CHANNEL_NAME**.

In this section, we list the information Televic Education will provide to you (as Identity Provider) and also need to receive from you in order to successfully setup the SSO between both platforms.

4.1 Information provided by Televic Education

In order to configure assessmentQ as a client in the Identity Provider, we provide following connection details.

Relying party WS-Federation Passive protocol URL =

https://www.sign-in.education/e/sso/*<subdomain>* (PRODUCTION)
https://demo.sign-in.education/e/sso/*<subdomain>*(DEMO)

Relying party identifier =

urn:*<subdomain>*.sign-in.education (PRODUCTION)
urn:*<subdomain>*.demo.sign-in.education (DEMO)

Relying party identifier =

https://www.sign-in.education/e/.well-known/openid-configuration (PRODUCTION)
https://demo.sign-in.education/e/.well-known/openid-configuration (DEMO)

Note: subdomain is case sensitive and should always be lower case.

4.2 Information to provide by the Identity Provider

In order to configure the Identity Provider within assessmentQ, Televic Education requires following information from you.

ADFS Federation Metadata file URL =

ADFS Federation Metadata file URL

(Note: if needed, the ADFS Federation Metadata file can be provided directly instead of the URL)

Test account =

TEST ACCOUNT DETAILS *(in order to verify the SSO integration)*

Please note that for this test account, the first name, last name and e-mail address must be configured correctly as these are required fields for the integration.