# Azure ADFS SSO Integration

To enable single sign on using Azure ADFS, we need to configure both Azure ADFS and Incorta. This document captures all the steps needed.

An important thing to note that Azure ADFS accepts https URLs only , so incorta should be either https enable or any other solutions that make incorta urls are https

## Step-by-step guide

For each tenant that is needed to be SSO enabled , the following steps should be repeated.  
It should be noted that the following steps were done on premium azure account .

Azure ADFS Configurations :

* Select "Azure Active Directory" then "Enterprise Application"
* Add new application
* Select non gallery Application
* Select Single sign-on
* In Single Sign-on Mode, select  SAML-based sign-on
* In the "Identity (Entity ID)" add unique value in your tenant  for example [https://demo.incorta:8443/incorta](https://localhost:8443/incorta)
* In the "Reply URL" in Azure ADFS enter incorta installation URL
  + pattern is  [https://<incorta-server>/incorta/!<tenant-name>/](https://localhost:8443/incorta/!demo/)
* Check View and edit all other user attributes
* In the SAML Token Attributes section , Click on the Add attribute URL
* in the name field enter loginName and in the value  field select a relevant field (this value should be as the login name field in Incorta installation) and then click OK button
* Download Metadata xml (You will need while configuring incorta SSO property)
* Click on the configure section

## Incorta Configuration

On Incorta side, following configuration changes are needed:

1. Enable Single Sign on at Tenant Level
2. Create a Configuration file
3. Modify server.xml to refer configuration file created in step 2
4. Bounce Incorta

### Enable Single Sign on at Tenant Level

A tenant can be single sign on enabled either from Admin screen or the tmt command .

Using Admin Screen :

* Navigate to admin screen
* Click on Tenents tab
* Click on the desired tenant name

Using tmt command :

./tmt.sh --update-property <tenantname> sso-login-enable true

Example: ./tmt.sh --update-property demo sso-login-enable true

### Create a Configuration file

Create a configuration file on Incorta. Pasted below is a sample configuration file .

Change the following properties :

* onelogin.saml2.sp.entityid : Should be same value as in the "Identity (Entity ID)" as configured in Azure ADFS).
* onelogin.saml2.sp.assertion\_consumer\_service.url :Should be same value as "Reply URL" in Azure ADFS.
  + pattern is  [https://<incorta-server>/incorta/!<tenant-name>/](https://localhost:8443/incorta/!demo/)
* onelogin.saml2.sp.single\_logout\_service.url : Should be , "http:///<incortaHostName>/incorta/logout.jsp?rediredtUrl=." .
* onelogin.saml2.idp.entityid = Should be the same value entityID attribute in Azure ADFS metadata xml file (can be downloaded from "4. SAML Signing Certificate" section in the Metadata XML).
* onelogin.saml2.idp.single\_sign\_on\_service.url : Should be the same value in the "Location" attribute in the   
  "SingleSignOnService" tag in Azure ADFS metadata xml file
* onelogin.saml2.idp.single\_logout\_service.url = Should be <https://login.microsoftonline.com/common/wsfederation?wa=wsignout1.0> .
* onelogin.saml2.idp.x509cert :Should be the value of "X509Certificate" in Azure ADFS metadata xml file .

Fields that are highlighted in red are customer specific.

------ Beginning of the File   /home/incorta/IncortaAnalytics/sso/ssoDemoConf.properties     --------

# If 'strict' is True, then the Java Toolkit will reject unsigned  
# or unencrypted messages if it expects them signed or encrypted  
# Also will reject the messages if not strictly follow the SAML  
onelogin.saml2.strict = false

# Enable debug mode (to print errors)  
onelogin.saml2.debug = true

# Service Provider Data that we are deploying  
#v

# Identifier of the SP entity (must be a URI)  
onelogin.saml2.sp.entityid = <https://localhost:8443/incorta>

# Specifies info about where and how the <AuthnResponse> message MUST be  
# returned to the requester, in this case our SP.  
# URL Location where the <Response> from the IdP will be returned  
#onelogin.saml2.sp.assertion\_consumer\_service.url = <http://localhost:8080/java-saml-tookit-jspsample/acs.jsp>  
onelogin.saml2.sp.assertion\_consumer\_service.url = <https://localhost:8443/incorta/!demo/>

# SAML protocol binding to be used when returning the <Response>  
# message. Onelogin Toolkit supports for this endpoint the  
# HTTP-POST binding only  
onelogin.saml2.sp.assertion\_consumer\_service.binding = urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect

# Specifies info about where and how the <Logout Response> message MUST be  
# returned to the requester, in this case our SP.  
onelogin.saml2.sp.single\_logout\_service.url = <https://localhost:8443/incorta/logout.jsp?rediredtUrl=>.

# SAML protocol binding to be used when returning the <LogoutResponse> or sending the <LogoutRequest>  
# message. Onelogin Toolkit supports for this endpoint the  
# HTTP-Redirect binding only  
onelogin.saml2.sp.single\_logout\_service.binding = urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect

# Specifies constraints on the name identifier to be used to  
# represent the requested subject.  
# Take a look on lib/Saml2/Constants.php to see the NameIdFormat supported  
#onelogin.saml2.sp.nameidformat = urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified

# Usually x509cert and privateKey of the SP are provided by files placed at  
# the certs folder. But we can also provide them with the following parameters

onelogin.saml2.sp.x509cert =  
# Requires Format PKCS#8 BEGIN PRIVATE KEY   
# If you have PKCS#1 BEGIN RSA PRIVATE KEY convert it by openssl pkcs8 -topk8 -inform pem -nocrypt -in sp.rsa\_key -outform pem -out sp.pem  
onelogin.saml2.sp.privatekey =

# Identity Provider Data that we want connect with our SP  
#

# Identifier of the IdP entity (must be a URI)  
onelogin.saml2.idp.entityid =<https://sts.windows.net/e1641373-1717-4ca1-aac0-c1fafd043b16/>

# SSO endpoint info of the IdP. (Authentication Request protocol)  
# URL Target of the IdP where the SP will send the Authentication Request Message  
onelogin.saml2.idp.single\_sign\_on\_service.url = <https://login.microsoftonline.com/e1641373-1717-4ca1-aac0-c1fafd043b16/saml2>  
onelogin.saml2.security.want\_nameid = false  
# SAML protocol binding to be used when returning the <Response>  
# message. Onelogin Toolkit supports for this endpoint the  
# HTTP-Redirect binding only  
onelogin.saml2.idp.single\_sign\_on\_service.binding = urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect  
#if the above did not work try the below  
#onelogin.saml2.idp.single\_sign\_on\_service.binding = urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST

# SLO endpoint info of the IdP.  
# URL Location of the IdP where the SP will send the SLO Request  
#onelogin.saml2.idp.single\_logout\_service.url = <https://incorta-dev.onelogin.com/trust/saml2/http-redirect/slo/610260>  
onelogin.saml2.idp.single\_logout\_service.url = <https://login.microsoftonline.com/common/wsfederation?wa=wsignout1.0>  
# <https://login.microsoftonline.com/e1641373-1717-4ca1-aac0-c1fafd043b16/saml2>  
# Optional SLO Response endpoint info of the IdP.  
# URL Location of the IdP where the SP will send the SLO Response. If left blank, same URL as onelogin.saml2.idp.single\_logout\_service.url will be used.  
# Some IdPs use a separate URL for sending a logout request and response, use this property to set the separate response url  
onelogin.saml2.idp.single\_logout\_service.response.url =

# SAML protocol binding to be used when returning the <Response>  
# message. Onelogin Toolkit supports for this endpoint the  
# HTTP-Redirect binding only  
onelogin.saml2.idp.single\_logout\_service.binding = urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect

# Public x509 certificate of the IdP  
onelogin.saml2.idp.x509cert =

# Instead of use the whole x509cert you can use a fingerprint  
# (openssl x509 -noout -fingerprint -in "idp.crt" to generate it,  
# or add for example the -sha256 , -sha384 or -sha512 parameter)  
#  
# If a fingerprint is provided, then the certFingerprintAlgorithm is required in order to  
# let the toolkit know which Algorithm was used. Possible values: sha1, sha256, sha384 or sha512  
# 'sha1' is the default value.  
# onelogin.saml2.idp.certfingerprint =   
# onelogin.saml2.idp.certfingerprint\_algorithm = sha1

# Security settings  
#

# Indicates that the nameID of the <samlp:logoutRequest> sent by this SP  
# will be encrypted.  
onelogin.saml2.security.nameid\_encrypted = false

# Indicates whether the <samlp:AuthnRequest> messages sent by this SP  
# will be signed. [The Metadata of the SP will offer this info]  
onelogin.saml2.security.authnrequest\_signed = false

# Indicates whether the <samlp:logoutRequest> messages sent by this SP  
# will be signed.  
onelogin.saml2.security.logoutrequest\_signed = false

# Indicates whether the <samlp:logoutResponse> messages sent by this SP  
# will be signed.  
onelogin.saml2.security.logoutresponse\_signed = false

# Sign the Metadata  
# Empty means no signature, or comma separate the keyFileName and the certFileName  
onelogin.saml2.security.want\_messages\_signed =

# Indicates a requirement for the <samlp:Response>, <samlp:LogoutRequest> and  
# <samlp:LogoutResponse> elements received by this SP to be signed.  
onelogin.saml2.security.want\_assertions\_signed = false

# Indicates a requirement for the Metadata of this SP to be signed.  
# Right now supported null (in order to not sign) or true (sign using SP private key)   
onelogin.saml2.security.sign\_metadata = false

# Indicates a requirement for the Assertions received by this SP to be encrypted  
onelogin.saml2.security.want\_assertions\_encrypted = false

# Indicates a requirement for the NameID received by this SP to be encrypted  
onelogin.saml2.security.want\_nameid\_encrypted = false

# Authentication context.  
# Set Empty and no AuthContext will be sent in the AuthNRequest,  
# Set comma separated values urn:oasis:names:tc:SAML:2.0:ac:classes:urn:oasis:names:tc:SAML:2.0:ac:classes:Password  
#onelogin.saml2.security.requested\_authncontext = urn:oasis:names:tc:SAML:2.0:ac:classes:urn:oasis:names:tc:SAML:2.0:ac:classes:Password  
#onelogin.saml2.security.requested\_authncontext = urn:oasis:names:tc:SAML:2.0:ac:classes:urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified,urn:oasis:names:tc:SAML:2.0:ac:classes:urn:oasis:names:tc:SAML:2.0:ac:classes:Password,urn:oasis:names:tc:SAML:2.0:ac:classes:urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport,urn:oasis:names:tc:SAML:2.0:ac:classes:urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient,urn:oasis:names:tc:SAML:2.0:ac:classes:urn:oasis:names:tc:SAML:2.0:ac:classes:X509,urn:oasis:names:tc:SAML:2.0:ac:classes:urn:federation:authentication:windows,urn:oasis:names:tc:SAML:2.0:ac:classes:urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos  
#onelogin.saml2.security.requested\_authncontext = urn:oasis:names:tc:SAML:2.0:ac:classes:urn:oasis:names:tc:SAML:2.0:ac:classes:Password,urn:oasis:names:tc:SAML:2.0:ac:classes:urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport,urn:oasis:names:tc:SAML:2.0:ac:classes:urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient,urn:oasis:names:tc:SAML:2.0:ac:classes:urn:oasis:names:tc:SAML:2.0:ac:classes:X509,urn:oasis:names:tc:SAML:2.0:ac:classes:urn:federation:authentication:windows,urn:oasis:names:tc:SAML:2.0:ac:classes:urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos  
onelogin.saml2.security.requested\_authncontext = urn:oasis:names:tc:SAML:2.0:ac:classes:Password  
# Allows the authn comparison parameter to be set, defaults to 'exact'  
#onelogin.saml2.security.requested\_authncontextcomparison = exact  
onelogin.saml2.security.requested\_authncontextcomparison = exact

# Indicates if the SP will validate all received xmls.  
# (In order to validate the xml, 'strict' and 'wantXMLValidation' must be true).  
onelogin.saml2.security.want\_xml\_validation = true

# Algorithm that the toolkit will use on signing process. Options:  
# '<http://www.w3.org/2000/09/xmldsig#rsa-sha1>'  
# '<http://www.w3.org/2000/09/xmldsig#dsa-sha1>'  
# '<http://www.w3.org/2001/04/xmldsig-more#rsa-sha256>'  
# '<http://www.w3.org/2001/04/xmldsig-more#rsa-sha384>'  
# '<http://www.w3.org/2001/04/xmldsig-more#rsa-sha512>'  
onelogin.saml2.security.signature\_algorithm = <http://www.w3.org/2000/09/xmldsig#rsa-sha1>

# Organization  
[onelogin.saml2.organization.name](http://onelogin.saml2.organization.name) = SP Java   
onelogin.saml2.organization.displayname = SP Java Example  
onelogin.saml2.organization.url = <http://sp.example.com>

# Contacts  
onelogin.saml2.contacts.technical.given\_name = Technical Guy  
onelogin.saml2.contacts.technical.email\_address = [technical@example.com](mailto:technical@example.com)  
onelogin.saml2.contacts.support.given\_name = Support Guy  
onelogin.saml2.contacts.support.email\_address = support@@[example.com](http://example.com)

                                          --------- End of File ------------

### Modify server.xml

Modify server.xml (located in <incorta home>/server/Conf/server.xml) to add the following tag at the beginning of <Host> tag:

 ```

 <Valve className="com.incorta.sso.valves.OneLoginValve"

 confFilesMap="Tenant\_Name=Absolute\_Path,Tenant\_Name2=Absolute\_Path2"

 LoggingEnabled = "true"

 />

 ```

   Where:

   1. Tenant\_Name is the name of Incorta Tenant.

   2. Absolute\_Path is the path of sso configuration file.

   3.  LoggingEnabled  This flag turns on the the valve logging messages. By default it's false which mean the logging is turned off.

Example  :

<Valve className="com.incorta.sso.valves.OneLoginValve"

 confFilesMap="demo=/home/incorta/IncortaAnalytics/sso/ssoDemoConf.properties"

 LoggingEnabled = "true"

 />

## Related articles

* Page:

[Inspector Executor](https://incorta.atlassian.net/wiki/wiki/spaces/INC/pages/4726751247/Inspector+Executor)

* Page:

[Automatic Table Selection](https://incorta.atlassian.net/wiki/wiki/spaces/INC/pages/4473487372/Automatic+Table+Selection)

* Page:

[How To Change The Thread Pool Used for ETL Steps](https://incorta.atlassian.net/wiki/wiki/spaces/INC/pages/4656824321/How+To+Change+The+Thread+Pool+Used+for+ETL+Steps)

* Page:

[Geminga](https://incorta.atlassian.net/wiki/wiki/spaces/INC/pages/4575166466/Geminga)

* Page:

[Supporting Multi Source data](https://incorta.atlassian.net/wiki/wiki/spaces/INC/pages/3164241946/Supporting+Multi+Source+data)