Connecting a service

Get a SAML 2.0 Service Provider or Open ID Connect -product or library, and integrate it with your services authentication management systems.

MPASSid supports two alternative user authentication protocols, SAML2 and OpenID Connect "OIDC", from which the service provider may choose one that suits them best. Both protocols have mostly the same features and provide the same user information. The differences are in the technical details.

The service must support the MPASSid SAML 2.0 or OIDC Profile.

Modifying the service to be compatible with MPASSid requires a certain knowledge of either SAML2 or OIDC -protocols, and the libraries or software that implement them.

MPASSid's technical support focuses on the API's message exchange, but cannot provide support with basic protocol implementation.

Good resources for the basics:

- SAML V2.0 Executive Overview
- https://wiki.oasis-open.org/security/FrontPage - SAML Wiki
- https://openid.net/connect/ - OIDC homepage

Please note! If you are using OIDC -protocol: MPASSid strongly suggests that you only use products/libraries certified by the OpenID Foundation, to ensure a secure and standard OIDC-support.

Before joining the MPASSid trust network and providing service, it's best to test the service with MPASSid. For this, MPASSid-operator provides a testing environment to which the service provider may integrate for testing. In the test environment, you can authenticate to a service with test logins and test how MPASSid provides the user information. The test environment does not process real personal data.

Integration with SAML2-protocol

Integrating a service to MPASSid's with SAML2-protocol requires the service provider to submit service SAML2 technical information, the saml2-metadata, to the MPASSid-operator. The service provider must also add MPASSid's identity provider metadata as a trusted service. After metadata exchange, MPASSid and the service can establish a trust relationship, and transfer data and communicate securely. All the information must be convergent and correct at all times.

Information to deliver to MPASSid:

- SAML2-services entity ID, which is the service's technical unique name. Best practice to ensure uniqueness is using the name format https://domain.fi/service
- AssertionConsumerService (ACS) URL address where MPASSid redirects authentication responses.
- Signing certificate, which is used to sign SAML Requests.
- Encryption certificate if assertions encryption is used. Encryption of assertions is recommended.

Collecting and submitting this information requires the service to have proper configurations for the SAML2 service provider role. The information is sent to the MPASSid's support by email at mpass@oph.fi

After the trust relationship has been established, all authentication requests are directed to the SingleSignOnService -URL defined in MPASSid's metadata. MPASSid uses the saml2int -profile, which means that the authentication requests must be sent with the browser by the HTTP-Redirect -binding, and the authentication responses are sent back with the HTTP-POST -binding. After a proper authentication request, the user can authenticate at the MPASSid. After authentication, the user is redirected back to the service, with MPASSid's response message, which contains the user information.

Testing with SAML2-protocol

MPASSid test environment's SAML2-metadata can be found at https://mpass-proxy-test.csc.fi/idp/shibboleth

In the test enviroment AssertionConsumerService URL can also be localhost url's.

Instructions for test login can be found below.

Integration with OpendID Connect -protocol

The Integration of service with OIDC requires the service provider to submit a list of the services OIDC redirect_uri -addresses to the MPASSid-operator. By default, services are expected to use the **authorization code flow** and the **client_secret_basic** -method when authenticating to the OIDC token endpoint, but other standard methods are also supported. This information (for example public key information, when private_key_jwt -method is being used) is submitted to the operator. In return, MPASSid-operator will send the client_id -value, and other attributes required by the chosen profile, such as the client_secret. Information is sent to the MPASSid support by email at mpass@oph.fi .

After registering the service, authentication requests are directed to the authorization_endpoint defined in MPASSid’s OpenID-configuration. Details about the redirect are dependent on the used OIDC-profile/flow: MPASSid supports all flows defined in the standard. After a proper authentication request, the user can authenticate at the MPASSid. Once authenticated, the user is redirected to the service with MPASSid’s response message, which contains either the user information or information needed to receive the user information from the OIDC token_endpoint, depending on the used OIDC flow.

Testing with OIDC

MPASSid test environments OIDC-information can be found at https://mpass-proxy-test.csc.fi/.well-known/openid-configuration

In the test enviroment redirect_uri's can also be localhost url's and https is not required.

Instructions for test login can be found below.

Production

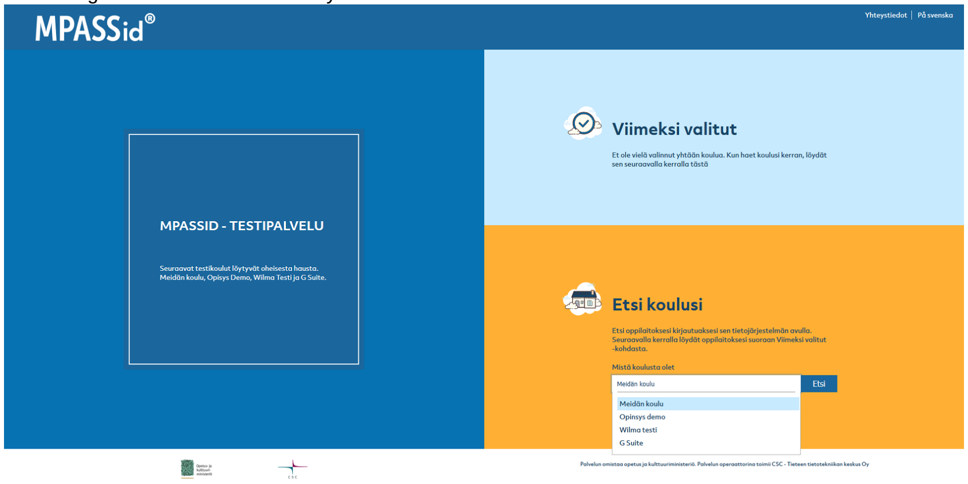
Before deploying the service to the production, the service provider must sign MPASSid’s member agreement and test the service in the test environment. The service is taken into production by submitting the same information about the service's production environment as from the test environment to MPASSid’s support at mpass@oph.fi

MPASSid’s production environments SAML2-metadata can be found at https://mpass-proxy.csc.fi/idp/shibboleth and OIDC-information at https://mpass-proxy.csc.fi/.well-known/openid-configuration.

Authenticating at the test environment

After a proper authentication request, the user’s browser is redirected to MPASSid’s test-environment.

- For testing select “Meidän koulu” as your school.



MPASSID - TESTIPALVELU

Seuraavat testikoulut löytyvät oheisesta hausta. Meidän koulu, Opisys Demo, Wilma Testi ja G Suite.

Viimeksi valitut

Et ole vielä valinnut yhtään koulua. Kun haet koulusi kerran, löydät sen seuraavalla kerralla tästä

Etsi koulusi

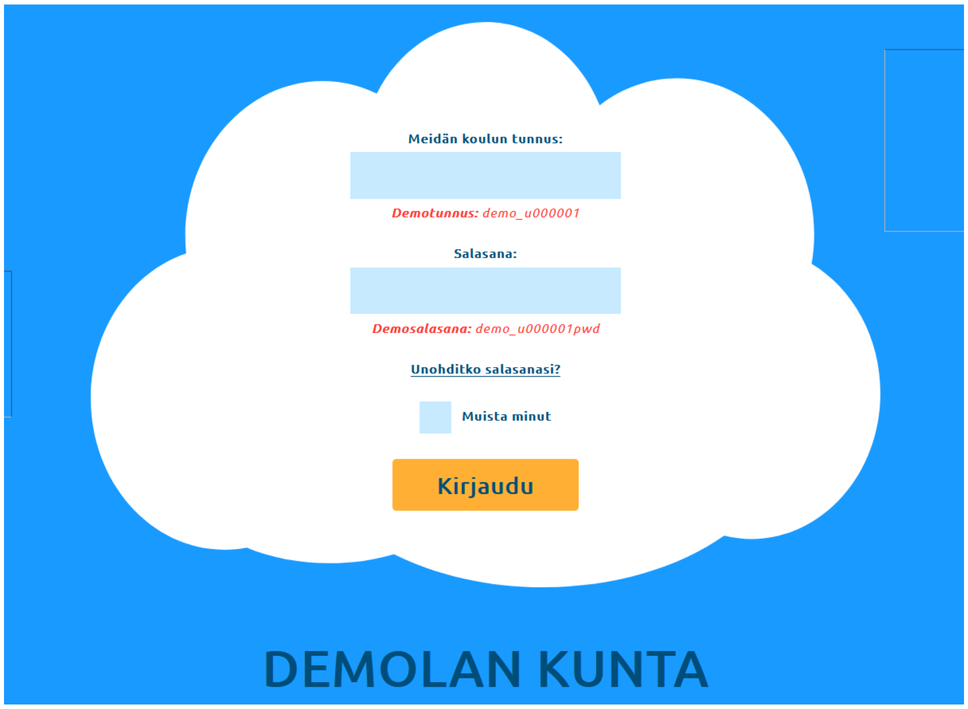
Etsi oppilaitoksesi kirjautuaksesi sen tietojärjestelmän avulla. Seuraavalla kerralla löydät oppilaitoksesi suoraan Viimeksi valitut -kohdasta.

Mistä koulusta olet: Meidän koulu

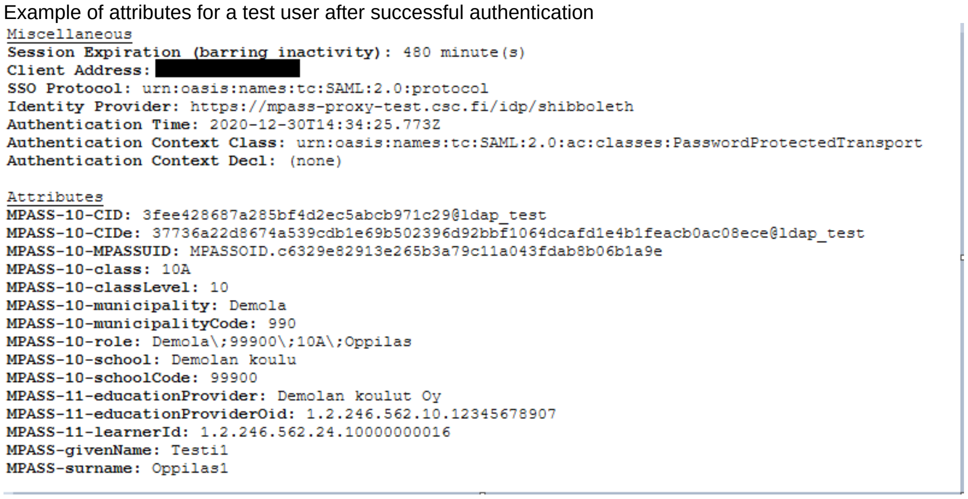
Meidän koulu
Opinsys demo
Wilma testi
G Suite

Palvelun omistaa opetus ja kulttuuriministeriö. Palvelun operaattorina toimii CSC - Tieteen tietotekniikan keskus Oy

- Log in with the user credentials available on the screen, or choose one of the test-users for testing with different users Test accounts available for testing



- Example of attributes for a test user after successful authentication



```
Miscellaneous
Session Expiration (barring inactivity): 480 minute(s)
Client Address: [redacted]
SSO Protocol: urn:oasis:names:tc:SAML:2.0:protocol
Identity Provider: https://mpass-proxy-test.csc.fi/idp/shibboleth
Authentication Time: 2020-12-30T14:34:25.773Z
Authentication Context Class: urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
Authentication Context Decl: (none)

Attributes
MPASS-10-CID: 3fee428687a285bf4d2ec5abcb971c29@ldap_test
MPASS-10-CIDe: 37736a22d8674a539cdb1e69b502396d92bbf1064dcafd1e4b1feacb0ac08ece@ldap_test
MPASS-10-MPASSUID: MPASSOID.c6329e82913e265b3a79c11a043fdab8b06b1a9e
MPASS-10-class: 10A
MPASS-10-classLevel: 10
MPASS-10-municipality: Demola
MPASS-10-municipalityCode: 990
MPASS-10-role: Demola\;99900\;10A\;Oppilas
MPASS-10-school: Demolan koulu
MPASS-10-schoolCode: 99900
MPASS-11-educationProvider: Demolan koulut Oy
MPASS-11-educationProviderOid: 1.2.246.562.10.12345678907
MPASS-11-learnerId: 1.2.246.562.24.10000000016
MPASS-givenName: Testi1
MPASS-surname: Oppilas1
```

Test accounts available for testing

You can use test accounts found at Test accounts available for testing

Examples of multivalue attributes

Please note that MPASSid cannot guarantee any specific order in which the different values of a multivalue attribute are delivered. The specification document for SAML2 states that implementations must not depend on specific sorting orders for values (see lines 297-299 in the SAML2 standard protocols). Thus, value orders may differ in integrations from different home organisations.

```
</saml2:Attribute>
<saml2:Attribute Name="urn:mpass.id:role"
                 NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                 >
    <saml2:AttributeValue>Demola;99900;5B;Oppilas</saml2:AttributeValue>
    <saml2:AttributeValue>Demola;99901;5B;Oppilas</saml2:AttributeValue>
</saml2:Attribute>
<saml2:Attribute FriendlyName="sn"
                 Name="urn:oid:2.5.4.4"
                 NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                 >
    <saml2:AttributeValue>Oppilas91</saml2:AttributeValue>
</saml2:Attribute>
<saml2:Attribute FriendlyName="municipalityCode"
                 Name="urn:mpass.id:municipalityCode"
                 NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                 >
    <saml2:AttributeValue>990</saml2:AttributeValue>
</saml2:Attribute>
<saml2:Attribute Name="urn:mpass.id:schoolCode"
                 NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                 >
    <saml2:AttributeValue>99901</saml2:AttributeValue>
    <saml2:AttributeValue>99900</saml2:AttributeValue>
</saml2:Attribute>
<saml2:Attribute Name="urn:mpass.id:uid"
                 NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                 >
    <saml2:AttributeValue>MPASSOID.2f9134d33dde065765ffa5dc60842c66fb8b82e4</saml2:AttributeValue>
</saml2:Attribute>
<saml2:Attribute Name="urn:mpass.id:class"
                 NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                 >
    <saml2:AttributeValue>5B</saml2:AttributeValue>
</saml2:Attribute>
<saml2:Attribute FriendlyName="givenName"
                 Name="urn:oid:2.5.4.42"
                 NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                 >
    <saml2:AttributeValue>Testi91</saml2:AttributeValue>
</saml2:Attribute>
```