Create your own WAYF

To have more control over the IdP Discovery, you can create your own WAYF selection page and integrate it in your service. It goes as follows:

1. Be sure that your service is configured in SURFconext (Production or Test environment).
2. Request the file **"**SURFconext IdPs metadata**"** at https://metadata(.test).surfconext.nl/
    a. This contains all IdPs on the platform. You can also request a version that is scoped to your Service Provider: that only lists the IdPs connected to your SP, and will be updated automatically when this list changes. Request that from support@surfconext.nl.
3. Configure the metadata into your Service Provider directly or use it (e.g. with Xpath) to extract the Display names and SSO locations of the IdPs to be shown on your WAYF.
   The SSO location points to a SURFconext endpoint and has a specific identifier at the end. This identifier enables SURFconext to forward the authentication request to the requested IdP.

Example of a metadata file with SAML

```
<?xml version="1.0"?>
<md:EntitiesDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" xmlns:mdui="urn:oasis:names:tc:SAML:
2.0:metadata:ui" ID="CORTO6d017189c6bcd01c19935006ce6b32e89e29b4a3"><ds:Signature xmlns:ds="http://www.w3.org
/2000/09/xmldsig#"><ds:SignedInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#"><ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/><ds:SignatureMethod Algorithm="http://www.w3.org/2000/09
/xmldsig#rsa-sha1"/><ds:Reference URI="#CORTO6d017189c6bcd01c19935006ce6b32e89e29b4a3"><ds:Transforms><ds:
Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/><ds:Transform Algorithm="
http://www.w3.org/2001/10/xml-exc-c14n#"/></ds:Transforms><ds:DigestMethod Algorithm="http://www.w3.org/2000
/09/xmldsig#sha1"/><ds:DigestValue>L8ANkHPH4msXsIUFptAMeNTuMzQ=</ds:DigestValue></ds:Reference></ds:
SignedInfo><ds:SignatureValue>kSE6aUY74Y1P/B6ZDY4s6F3AEpCv0t
/z9fyyhUmPZctfshkiyK53vz8lKfmgiUlOk2c4+dXVPlVQqzeVgW2lDKycdWhkjSQnybBNPrBYlvlEPMJHO4p83IEOMGXh7yS6a8OjNc9qLTi
kVQnxwfV3xAZGxZ0AZVSJM9WhkqRMJGAK7xMcttM77cIy06ZRpNDb5e36Fb6dLHHAJ3JICd9CEHqdP3WKB2rO2wDGxrkIx
/6ynnM1YCFbWvpGU+dGT6/r7YTU9q89UdU2cYMTP1t4KSl/BOMJflnwlAEmFxcxn4FGKny9cRpzhu0nvmtk02cK8T
/pYboWWEqG6ooTIEM3Yw==</ds:SignatureValue><ds:KeyInfo><ds:X509Data><ds:X509Certificate></ds:X509Certificate><
/ds:X509Data></ds:KeyInfo></ds:Signature>
<md:EntityDescriptor validUntil="2012-05-31T22:00:00Z" entityID="https://test.test.nl"><md:SPSSODescriptor
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"><md:NameIDFormat>urn:oasis:names:tc:SAML:
2.0:nameid-format:transient</md:NameIDFormat><md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:
2.0:bindings:HTTP-POST" Location="https://engine.surfconext.nl/authentication/sp/consume-assertion
/7f301d787aa6ea235a8b86434d39aa41" index="1"/></md:SPSSODescriptor></md:EntityDescriptor>
<md:EntityDescriptor validUntil="2012-06-01T10:36:28Z" entityID="http://www.surf.nl/test"><md:
IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"><md:Extensions><mdui:
DisplayName xml:lang="en">SURFnet BV - This IdP is for testing only</mdui:DisplayName><mdui:DisplayName xml:
lang="nl">SURFnet BV - This IdP is for testing only</mdui:DisplayName><mdui:Description xml:lang="en"
>SURFnet BV - This IdP is for testing only</mdui:Description><mdui:Description xml:lang="nl">SURFnet BV -
This IdP is for testing only</mdui:Description><mdui:Logo height="60" width="120">https://wayf.surf.nl
/federate/surfnet/img/logo/surfnet.png</mdui:Logo><mdui:Keywords xml:lang="en">SURFNET</mdui:Keywords><mdui:
Keywords xml:lang="nl">SURFNET</mdui:Keywords></md:Extensions><md:KeyDescriptor xmlns:ds="http://www.w3.org
/2000/09/xmldsig#" use="signing"><ds:KeyInfo><ds:X509Data><ds:X509Certificate></ds:X509Certificate></ds:
X509Data></ds:KeyInfo></md:KeyDescriptor><md:KeyDescriptor xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
use="encryption"><ds:KeyInfo><ds:X509Data><ds:X509Certificate></ds:X509Certificate></ds:X509Data></ds:
KeyInfo></md:KeyDescriptor><md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</md:
NameIDFormat><md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect" Location="
https://engine.surfconext.nl/authentication/idp/single-sign-on/dedd75c2157a751113666d7888b2f2cd"/></md:
IDPSSODescriptor></md:EntityDescriptor>
```

Line 3 contains the Service Provider metadata of 'https://test.test.nl', line 4 the metadata for a coupled Identity Provider (SURFnet Test IdP). For simplicity the remainder of the XML metadata is omitted.

Shibboleth

When you use Shibboleth, the following HTML code presents a link that will start an authentication to a specific IdP:

```
<ul class="IdPlist">
<li><a
href="https://SP.example.org/Shibboleth.sso/Login?target=dashboard.php&entityID=http://www.surf.nl/test
">SURFconext Login</a></li>
</ul>
```

- The base URL is the URL of your Shibboleth Service Provider.
- 'target' contains the location to return to after the login was successful.
- entityID' is the EntityID of the IdP (as found in the SURFconext metadata).

Just expand the list with more IdPs from the SURFconext metadata, and you have created your own WAYF selection page.

OpenID Connect

When you use OpenID Connect, you can create a custom WAYF using the "login_hint" query parameter when calling the authorize endpoint. You can add the IdP entityID as value for this parameter. The entityID's of the connected institutions can be found in the published SAML IdPs metadata. For test, this metadata can be found here: https://metadata.test.surfconext.nl/idps-metadata.xml . For production, you can find it here: https://metadata.surfconext.nl/idps-metadata.xml
Once you have extracted the IdP entityID, you can use it like this (here we do the authorize request for the IdP with entityID: http://mock-idp):

```
https://connect.test.surfconext.nl/oidc/authorize?login_hint=http%3A%2F%2Fmock-
idp&scope=openid&response_type=code&redirect_uri=https%3A%2F%2Foidc-playground.test.surfconext.nl%
2Fredirect&state=example&nonce=example&client_id=playground_client&response_mode=query
```

The oidc-playground can be used to test the login_hint parameter.