Identity Provider

Description

> An identity provider (abbreviated IdP or IDP) is a system entity that creates, maintains, and manages identity information for principals and also provides authentication services to relying applications within a federation or distributed network.

An Identity Provider is responsible for identifying users. Also, it is responsible for giving service providers information regarding the identified user.

Soffid allows you to configure different identity providers, you can choose the best option for you by selecting the IdP type:

- **Soffid IdP**: identifies the identity provider implemented by Soffid. Soffid IdP implements both OpenID-Connect and SAML.
- **External SAML IdP**: is used to identify providers not implemented by Soffid. For instance, it could be an ADFS (Active Directory Federation Services) or Shibboleth identity provider.
- **OpenID-Connect**: is used for third-party identity providers, like ADFS.
- **Facebook**: if you select that option, oAuth2 will be used to identify Facebook users. You will need to register Soffid as a Facebook application to use it.
- **Google**: if you select that option OpenID-Connect will be used to identify Google users. You will need to register Soffid as a Google application to use it.
- **LinkedIn**: if you select that option, oAuth2 will be used to identify LinkedIn users. You will need to register Soffid as a LinkedIn application to use it.

To create an identity provider, it is advisable to install a dedicated sync server. It can be configured as a proxy sync server as it does not need direct access to the Soffid database. Instead, it will connect to the main sync server to get users and federation information.

> For more information about how to configure a dedicated sync server, you can visit the Install Sync server page.

Standard attributes

The fields for each IdP type are detailed below:

Soffid IdP

Identification

- **publicID**: unique name to identify the identity provider. The name has to be the same as the Public ID of the Soffid Identity Provider agent.
- **Name**: friendly user name.
- **Organization**: company name of the external IdP.
- **Contact**: email address of the external IdP.

> It will be mandatory to create an Agent (Soffid Identity Provider).

Service Configuration

- **Metadata**: the Metadata for an Identity Provider defines how this Identity Provider delivers its service:
  - Which security algorithms does it support.
  - The public portion of it's signing and encrypting keys.
  - The SAML protocols do it support.
  - The URL of each SAML protocol endpoint.
  - Contact information.

The Metadata is the information that any application needs to use the IdP. That is an XML file that contains the public encryption keys and the services provided

Leave it blank as Soffid IdP will fulfill it for you.

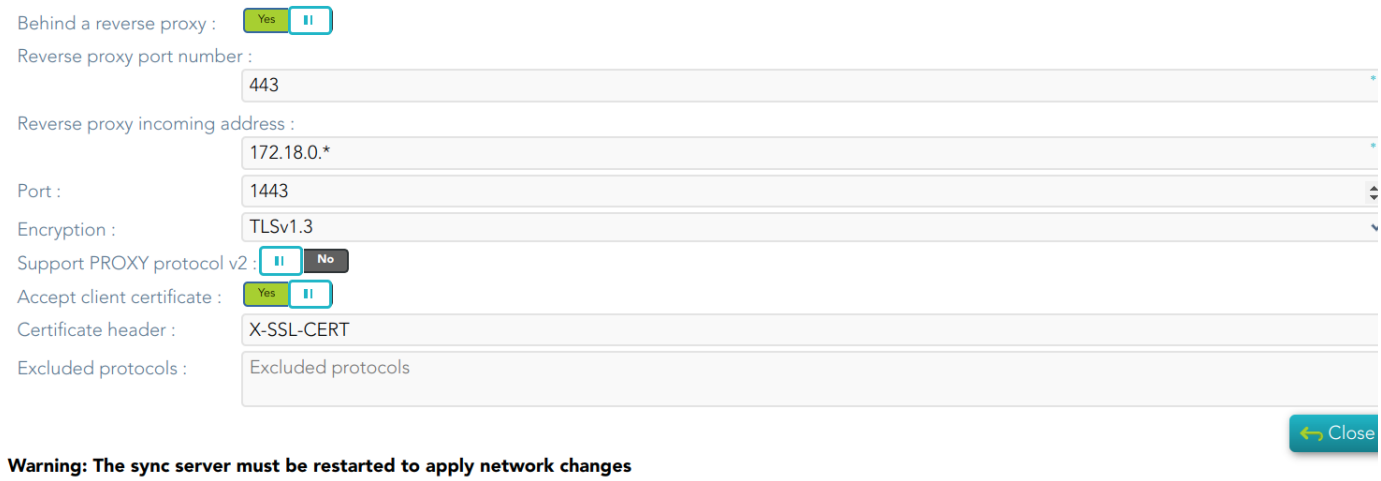
> The metadata will be created when the network data and SAML Security data. Restarting the sync server will be necessary to fill in the Metadata.

Network

- **Host name**: public hostname that will be used by users and service providers. The full qualified name should be used.
- **Allow IdP to be included inside an IFRAME**: Soffid allows you to configure the Identity Provider to be incluided within a IFRAME. If this option is updated, the Sync Server must be restarted. *This attribute will be available in Federation addon 3.5.37 or higher.*
- **Network ports**:
  - **Behind a reverse proxy**

- **Reverse proxy port number**: port where the reverse proxy is listening.
- **Reverse proxy incoming address**: IP addresses allowed to make calls to the reverse proxy.
- **Port**: TCP port number used by the identity provider. By default, TLS will be used (default 1443).
- **Encryption**: encryption type is only allowed behind a reverse proxy.
- **Support PROXY protocol v2**: protocol between the reverse proxy and the Identity Provider.
- **Accept client certificate**
- **Certificate header**: certificate data header (only behind a reverse proxy).
- **Excluded protocols**: encryption protocols to be excluded.

Image



- **TLS PublicKey**: there are three available options
  - **Leave in blank** and Soffid IdP will generate a self-signed certificate.
  - Clicking on the **Generates public/private key** button, a new private key pair will be generated. Once the private key pair is generated, you could generate a certificate request file, also known as PKCS#10 or CSR file. The certificate authority will be able to create a certificate for you using this certificate request. Once you have created the public/private key, you could run other new functions:
    - **Change public/private key**: allows you to change the public/private key generated previously.
    - **Delete public/private key**: allows you to delete the public/private key generated previously.
    - **Generate PKCS10**: generates a PKCS10 file (Certification request standard).
  - Clicking on the **Upload PKCS12 file** button it will be able to upload a PKCS#12 file. That file must contain the private and public keys and the server certificate as well. Mind that PKCS#12 file use to be protected by a PIN.
- **TLS Certificate chain**: text certificate chain created with one of the previous options.

> **Server certificate management:**there are two options for certificate management. You can visit the Server certificate management page for more information.

SAML Security

- **PublicKey**:
  - Clicking on the **Generates public / private key** button, a new private key pair will be generated. Once the private key pair is generated, you could generate a certificate request file, also known as PKC#10 or CSR file. The certificate authority will be able to create a certificate for you using this certificate request. Once you have created the public/private key, you could run other new functions:
    - **Change public/private key**: allows you to change the public/private key generated previously.
    - **Delete public/private key**: allows you to delete the public/private key generated previously.
    - **Generate PKCS10**: generates a PKCS10 file (Certification request standard).
  - Clicking on the **Upload PKCS12 file** button it will be able to upload a PKCS#12 file. That file must to contain the private an public keys and the server certificate as well. Mind that PKCS#12 file use to be protected by a PIN.
- **Certificate chain**: text certificate chain created with one of the previous options.

Session management

- **Session timeout (secs)**: time in seconds that will take the session. If the user has been authenticated, and later is requested to authenticate again, the user will be authenticated without any intervention as long as the timeout has not been elapsed.
- **oAuth Session timeout (secs)**: time in seconds that will take the oAuth session. The oAuth has its own life cycle, regardless the session timeout.
- **Maximum session duration (secs) :** maximum time during which session can be renewed
- **SSO Cookie name**: name of the cookie that will keep the session id, you can change the name. This SSO cookie is not really needed, as the identity provider will store a session cookie to track the SSO session. This SSO cookie is needed in two circumstances:
  - When the identity provider is restarted, the session cookie is lost. This SSO Cookie allows the identity provider to restart the lost session.
  - When you have more than one identity provider instance, this cookie allows all the identity providers to handle the session as if only was one identity provider. The SSO cookie can be allocated by any identity provider, and it will be accepted by any other one.
- **SSO Cookie domain**: is needed when you have more than one identity provider instance and they are using different host names. If all the identity providers are serving the same virtual host name, the SSO Cookie domain will be needed.

Authentication

- **Authentication methods**: matrix to define the authentication methods that will be required to successfully authenticate the user. Each row indicates the first authentication method, and each column indicates the second factor to use.
  - Password
  - Kerberos
  - External IdP
  - OTP
  - Email
  - SMS
  - PIN Certificate
  - FIDO
  - Push
- **Adaptive authentication**: that option allows you to add an additional authentication matrix which will be run when the condition defined was complied with. That is the way to change the authentication method depending on the environment.
  - **Description**: rule description to identify it.
  - **Condition**: script to enable that rule. The result of the rule must be true or false.

    There are some available vars to create the condition. You can visit the Condition for Adaptive authentication page for more information and some examples.
  - **Matrix**: to define the authentication methods that will be required to successfully authenticate the user. Each row indicates the first authentication method, and each column indicates the second factor to use.
- **Always ask for credentials:** if checked (the selected value is Yes), the IdP will always request credentials from users who meet the condition defined in this rule.
- **Register OTP when required:** if it is checked (selected value is Yes), Soffid will allow registering the OTP to users who meet the condition and do not have one previously.
- **Kerberos domain**: allows you to pick up a file to configure the Kerberos authentication method. For more information, you can visit the How to enable Kerberos authentication page.

Advanced Authentication

- **Allow user to recover password**: if it is checked (selected value is Yes), and the password recovery addon is installed, the user will be allowed to execute the password recovery mechanism.
- **Register OTP when required:** if it is checked (selected value is Yes), Soffid will allow to register the new OTP to the user during the login process.
- **Allow user to self-register**: if it is checked (selected value is Yes), the user will be allowed to register itself. This option sends an email to the user to verify the email address is correct, and then lets the user to enter a new password.
  - **Registration process:** workflow selected to create the new identity.
  - **User Type**: identifies the password policy that is to be applied. More information on this link User Type.
  - **Primary Group**: select which organization unit this user belongs to.

- **Register identities identified by external IdPs**: allows Soffid IdP to automatically register a new identity when a user authenticates with a third-party IdP, and this identity does not exist yet in Soffid database. Furthermore, at the third party IdP configuration page, one can tune how this identity is going to be created.
- **Store last user name in browser**: allows the browser to save the last user name when Yes is selected.
- **Enable reCaptcha v3 service**: *(*)* helps to keep save your website. You can enable it by selecting the Yes option. When you select the Yes option, you must fill in the following fields:
  - **Captcha site key**: this key is used to invoke the reCAPTCHA service
  - **Captcha site secret**: the secret key to communicate your web site with reCAPTCHA service. This secret key authorizes the communication.
  - **Captcha threshold (1 for highest confidence, 0 for low confidence)**:

Profiles

A profile is a protocol or subset of protocols implemented by the Identity Provider. There are some accepted protocols, those allows a custom config dependent on the selected profile.

> You can visit the Profiles chapter for more information about each one.

Look and feel

Soffid allows you to personalize your login page by adding some style elements, as well as header and footer elements.

- **Logo**: this logo will be displayed for user in Windows desktop.
- **CSS Style**: allows you to add a CSS style for your login page.
- **Html header**: allows you to add an Html header.
- **Html footer**: allows you to add an Html footer.
- **Language (2 characters code)**

External SAML IdP

Identification

- **publicID**: unique name to identify the identity provider.
- **Name**: friendly user name.
- **Organization**: company name of the external IdP.
- **Contact**: email address of the external IdP.

Service Configuration

- **Metadata**: the Metadata for an Identity Provider defines how this Identity Provider delivers its service:
  - Which security algorithms does it support.
  - The public portion of it's signing and encrypting keys.
  - The SAML protocols does it support.
  - The URL of each SAML protocol endpoint.
  - Contact information.

The Metadata is the information that any application need to use the IdP. That is an XML file that contains the public encryption keys and the services provided

Leave it blank as Soffid IdP will fulfill it for you.

Login Rules

- **User regular expression**: regular expression to detect users of this identity provider.
- **Login hint script**: script to help to login. Return the text to help.
- **Identity provisioning script**: script to bind or register a new identity. Return the user name of the owner identity for the authenticated account.

OpenID-Connect

Service Configuration

- **Metadata**: there are some required parameters:
  - **authorization_endpoint**: contains the oAuth endpoint to forward the user to get the authorization token.
  - **token_endpoint**: contains the oAuth endpoint to get the access token, based on the authorization token got at previous step.
  - **userinfo_endpoint**: if remote IdP is OpenID-connect compliant, the token endpoint should have sent an access token along a JWT OpenID token containing user claims. If this is not the case, Soffid will use this user_info endpoint to fetch user claims. This mechanism is needed for oAuth2 servers.
  - **scopes_sopported**: The list of scopes specified here will be used at first step, when redirecting the user to the authorization endpoint.

```
{
   "authorization_endpoint":
"https://server/oauth2/auth",
   "token_endpoint":
"https://server/oauth2/token",
   "userinfo_endpoint":
"https://server/oauth2/userinfo",
   "scopes_supported": [
"openid","email","profile"]
}
```

- **oAuth key**: is the identificator token generated by the oAuth server.
- **oAuth secret**: is the secret generated by the oAuth server.

The Metadata is the information that any application need to use the IdP. That is an XML file that contains the public encryption keys and the services provided

Login rules

- **User regular expression**: regular expression to detect users of this identity provider.
- **Login hint script**: script to help to login. Return the text to help.
- **Identity provisioning script**: script to bind or register a new identity. Return the user name of the owner identity for the authenticated account.

```
sn =
attributes{"screen_name"};
i = sn.indexOf(" ");
if (i> 0) {
 user.firstName = sn.substring(0,
i);
 user.lastName =
sn.substring(i+1);
} else {
 user.firstName = "?";
  user.lastName = sn;
}
return attributes{"name"};
```

Facebook

Identification

- **publicID**: unique name to identify the identity provider. Soffid will fulfill wint the Facebook URL.
- **Name**: friendly user name.
- **Organization**: company name of the external IdP.
- **Contact**: email address of the external IdP.

Service Configuration

- **Click here to obtain a client id and client secret**: allows you to get the oAuth key and secret.
- **oAuth key**: is the identificator token generated by the oAuth server.
- **oAuth secret**: is the secret generated by the oAuth server.

Login rules

- **User regular expression**: regular expression to detect users of this identity provider.
- **Login hint script**: script to help to login. Return the text to help.
- **Identity provisioning script**: script to bind or register a new identity. Return the user name of the owner identity for the authenticated account.

Google

Identification

- **publicID**: unique name to identify the identity provider. Soffid will fulfill wint the Google URL.
- **Name**: friendly user name.
- **Organization**: company name of the external IdP.
- **Contact**: email address of the external IdP.

Service Configuration

- **Click here to obtain a client id and client secret**: allows you to get the oAuth key and secret.
- **oAuth key**: is the identificator token generated by the oAuth server.
- **oAuth secret**: is the secret generated by the oAuth server.

Login rules

- **User regular expression**: regular expression to detect users of this identity provider.
- **Login hint script**: script to help to login. Return the text to help.
- **Identity provisioning script**: script to bind or register a new identity. Return the user name of the owner identity for the authenticated account.

Linkedin

Identification

- **publicID**: unique name to identify the identity provider. Soffid will fulfill wint the Linkedin URL.
- **Name**: friendly user name.
- **Organization**: company name of the external IdP.
- **Contact**: email address of the external IdP.

Service Configuration

- **Click here to obtain a client id and client secret**: allows you to get the oAuth key and secret.
- **oAuth key**: is the identificator token generated by the oAuth server.
- **oAuth secret**: is the secret generated by the oAuth server.

Login rules

- **User regular expression**: regular expression to detect users of this identity provider.
- **Login hint script**: script to help to login. Return the text to help.
- **Identity provisioning script**: script to bind or register a new identity. Return the user name of the owner identity for the authenticated account.

(*) What is CAPTCHA --> https://support.google.com/a/answer/1217728?hl=en

(*) https://www.google.com/recaptcha/about/

Revision #77
Created 8 September 2021 09:43:11
Updated 16 January 2026 15:56:42