# Identity providers (addon federation)

## Description

> This screen allows you to define the most important components of a federation, which are none other than the identity providers. An identity provider is responsible for performing the appropriate authentication for each service provider and user type according to their accounts, permissions, authorisations, and attributes.

The main supported standard is SAML. SAML allows to completely detach the identification process from web applications,  known as Service Providers. With SAML, identification is performed by specialized servers known as Identity Providers.  Additionaly, some other, less secure, but some times convenient protocols like OAuth (Open Authorization) and OpenID-Connect protocols are supported. Elder protocols like Openid (do not confuse with OpenID-Connect) are deprecated and no  longer supported.

Remember that after validating the user's login, the identity provider will send a set of attributes to the service provider that will have been previously defined in Soffid in the **attribute definition** page and **shared attribute policy** screens.

> You can visit the Introduction page to find more information about the federation.

> Please note that this screen is available in the federation addon.

## Entity group

> An entity group is just like a folder that allows you to manage different kinds of federation members. One of the most common ways to group federation members is by trust level.

When you create an entity group, identity provider records will be displayed.

Entity groups can be created on this screen or on the service provider screen, and they will be displayed on both screens.

# Identity provider

> 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.

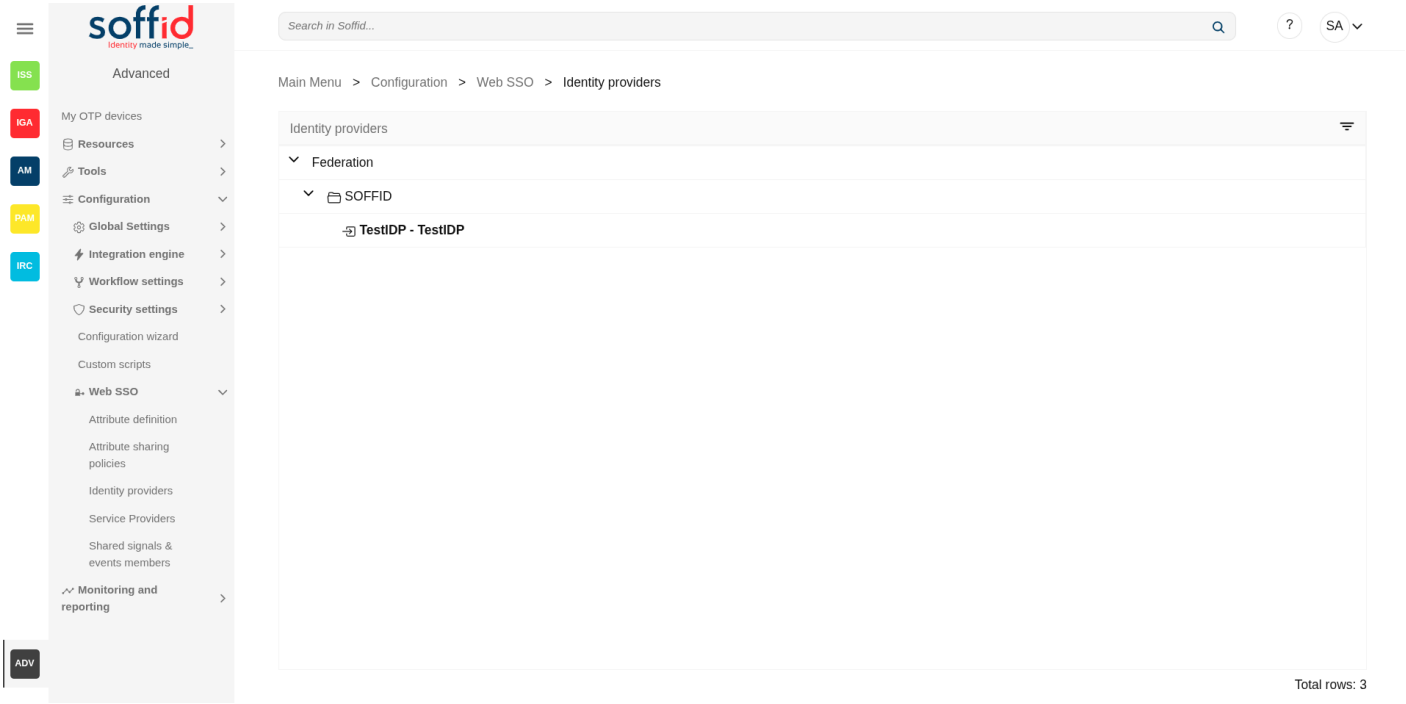
# Virtual identity provider

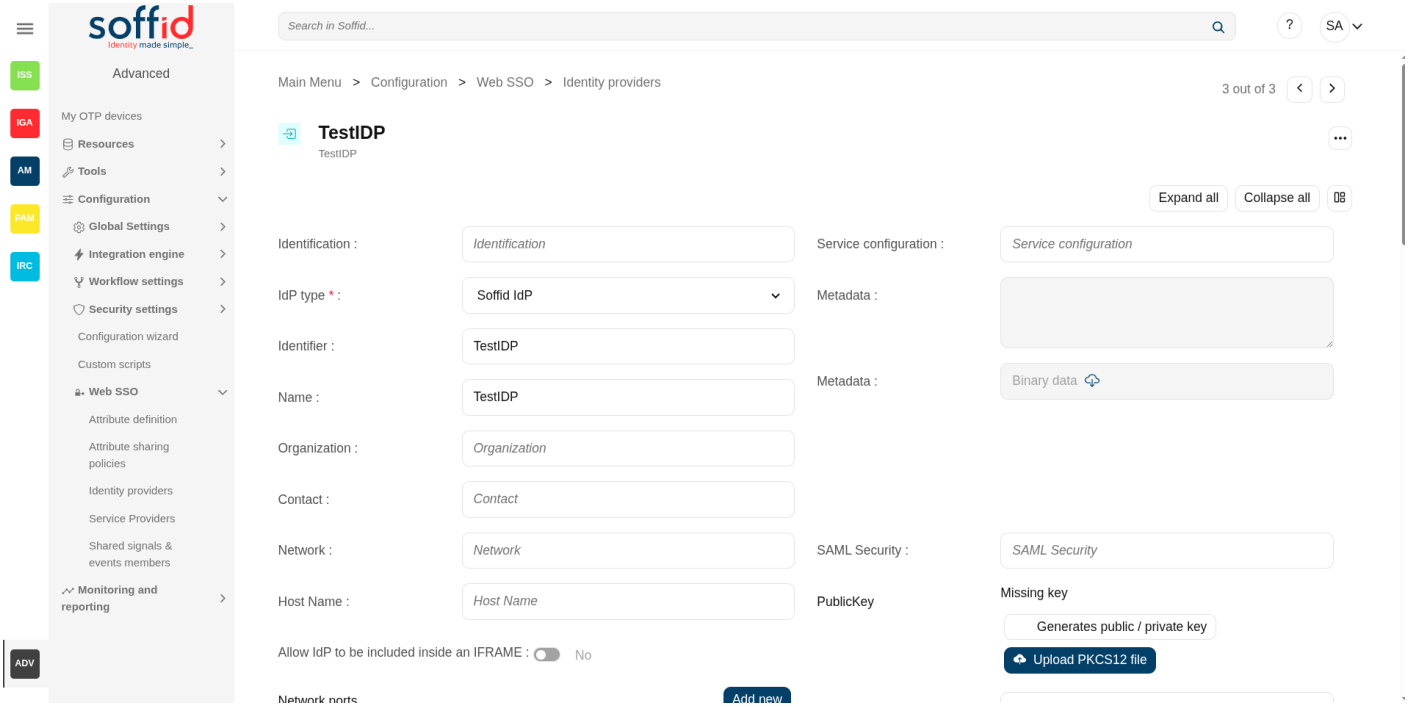
>

A single identity provider usually offers different profiles or service levels to diffeferent service provider. To be able to define this behavior, any Identity Provider can be split into many virtual identity providers. Those identity providers will be served by the same actual identity provider, but they will have different profile configurations.

When creating a new virtual identity provider, you will need to specify the service providers for which you will be responsible.

# Screen overview






# Related objects

- Attribute definition : where the list of possible attributes to be returned in the IdP response is defined
- Attribute sharing policies : where policies are defined with the attributes to be sent according to the authenticated service provider
- Identity providers : configuration of the identity providers
- Service providers : configuration of the service providers
- Metadata : where user attributes are defined

# Standard attributes

## Entity group

- **Entity Group**: name of the group.
- **Providers**: display the identity providers under the entigy group

## Identity provider

# Soffid IdP

## Identification

- **Idp type**: Soffid Idp (this one has to be selected)
- **Identifier**: 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) linking the idP with the identifier attribute.

## 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.
- **Metadata (file)**: from this field, you can directly download a file with the metadata.

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.

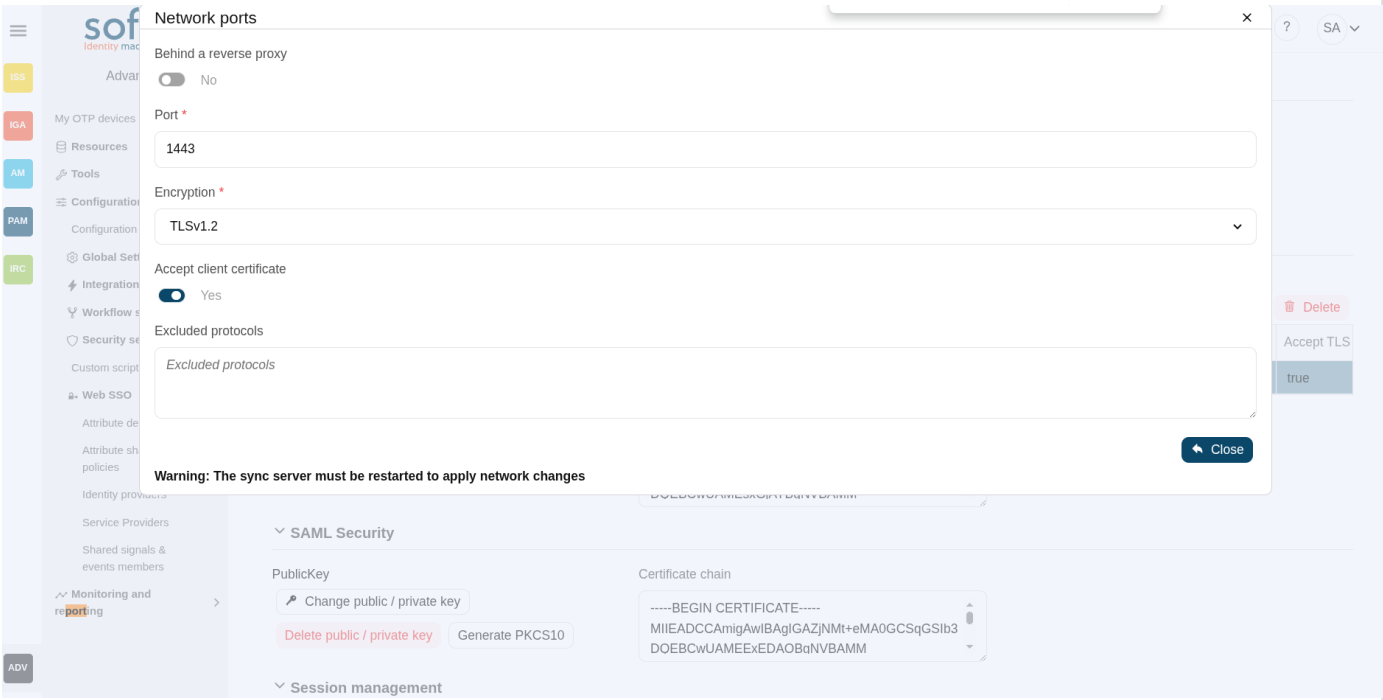
> The metadata will be created when the network data and SAML Security data are specified. Restarting the sync server will be necessary to fill in the Metadata.

## 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.
- **Network ports**:
  - **Behind a reverse proxy**: enable this option when the idp is behind a reverse proxy.

- **Reverse proxy port number**: (displayed when reverse proxy enabled) port where the reverse proxy is listening.
- **Reverse proxy incoming address**: (displayed when reverse proxy enabled) 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.
  - TLSv1.2
  - TLSv1.3
  - No encryption
- **Support PROXY protocol v2**: (displayed when reverse proxy enabled) protocol between the reverse proxy and the Identity Provider.
- **Accept client certificate**: to accept always the client certificate.
- **Certificate header**: (displayed when reverse proxy enabled) certificate data header.
- **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

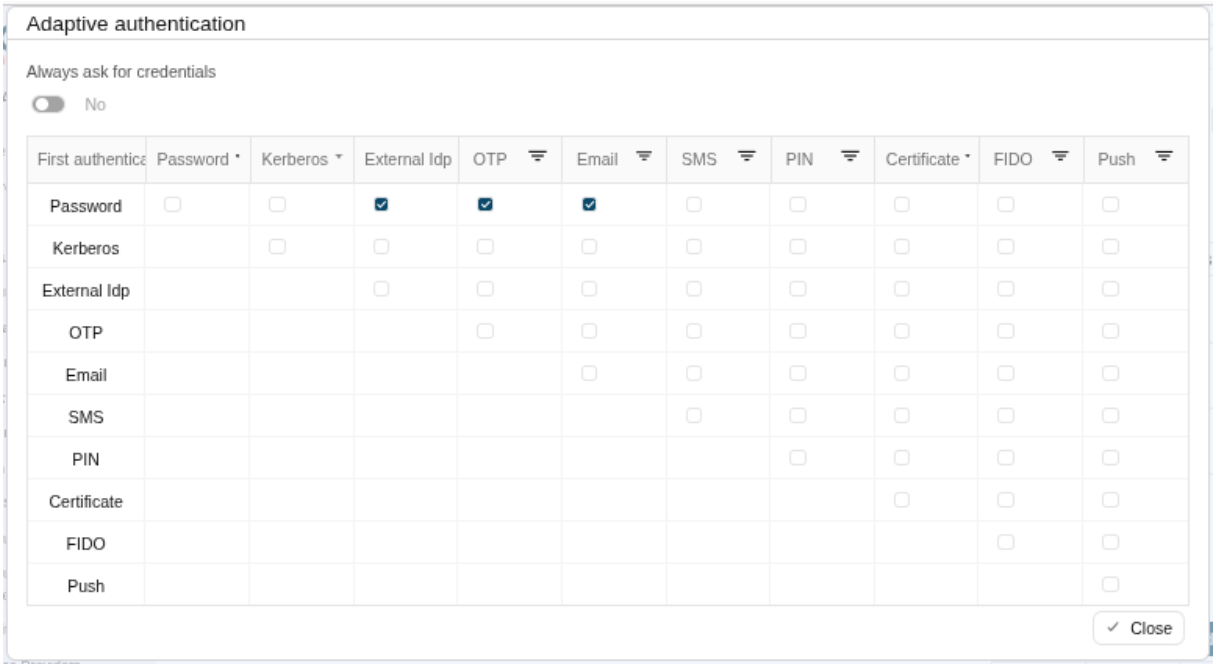
- **Session timeout (secs)**: time in seconds that will take the session. If the user has been authenticated, and later is requested to authenticate again, the user will be authenticated without any intervention as long as the timeout has not been elapsed.
- **oAuth Session timeout (secs)**: time in 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

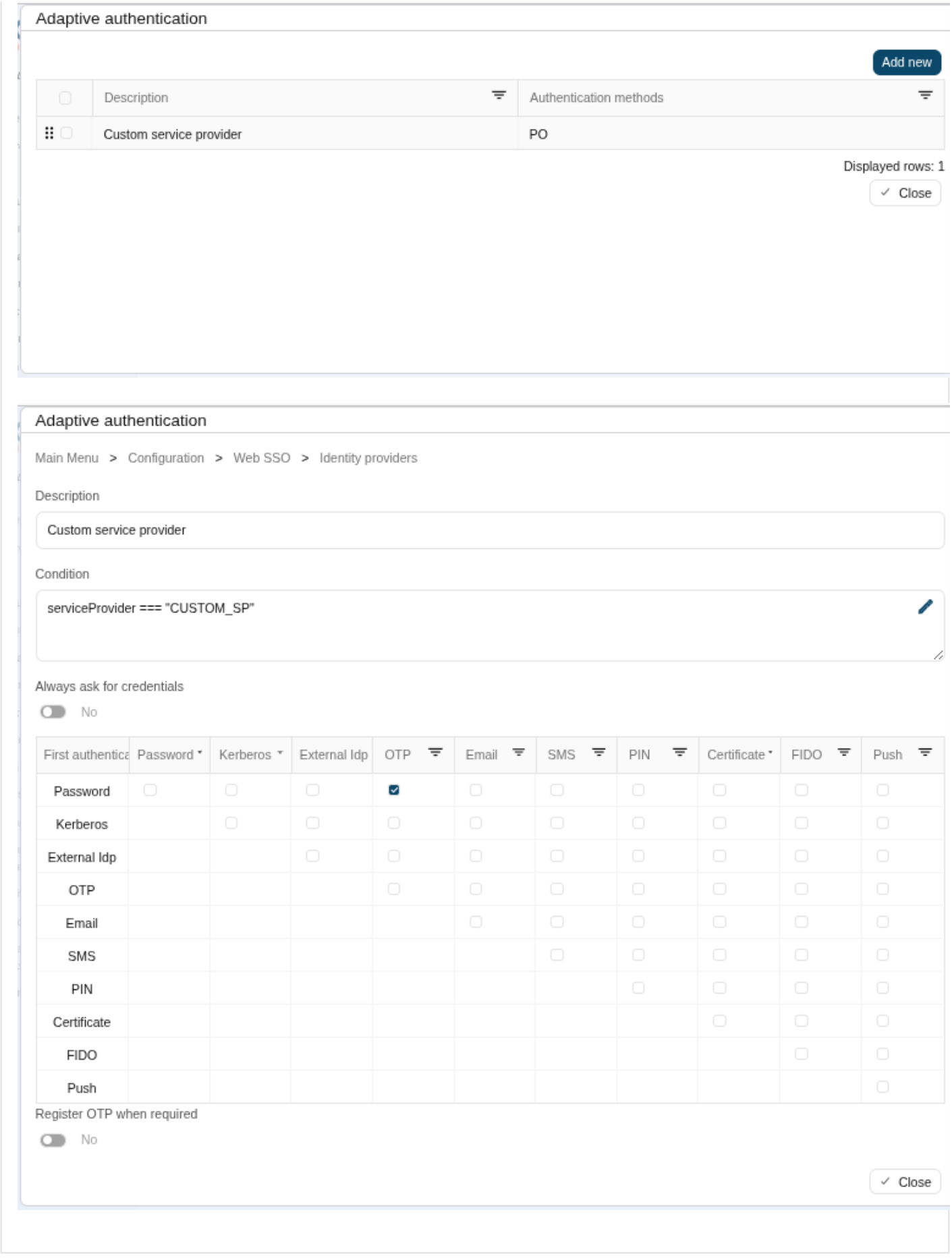
- **Default authentication methods**: the button open a popup.
    - **Always ask for credentials**: if checked (the selected value is Yes), the IdP will always request credentials from users who meet the condition defined in this rule.
    - **"Matrix of authentication methods"**: matrix to define the 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

| Image |
| --- |
| |




- **Adaptive authentication**: the button open a popup.
  - **"Table of adaptive authentication"**
    - **Description**: description of the adaptive authentication.
    - **Authentication methods**: displays the authentication methods seleccted.
  - **"Adaptive authentication popup"**: 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.
    - **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.
    - **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.

### Image




- **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**: (displayed when Allow users to self-service enabled) identifies the password policy that is to be applied. More information on this link User Type.
- **Primary Group**: (displayed when Allow users to self-service enabled)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.

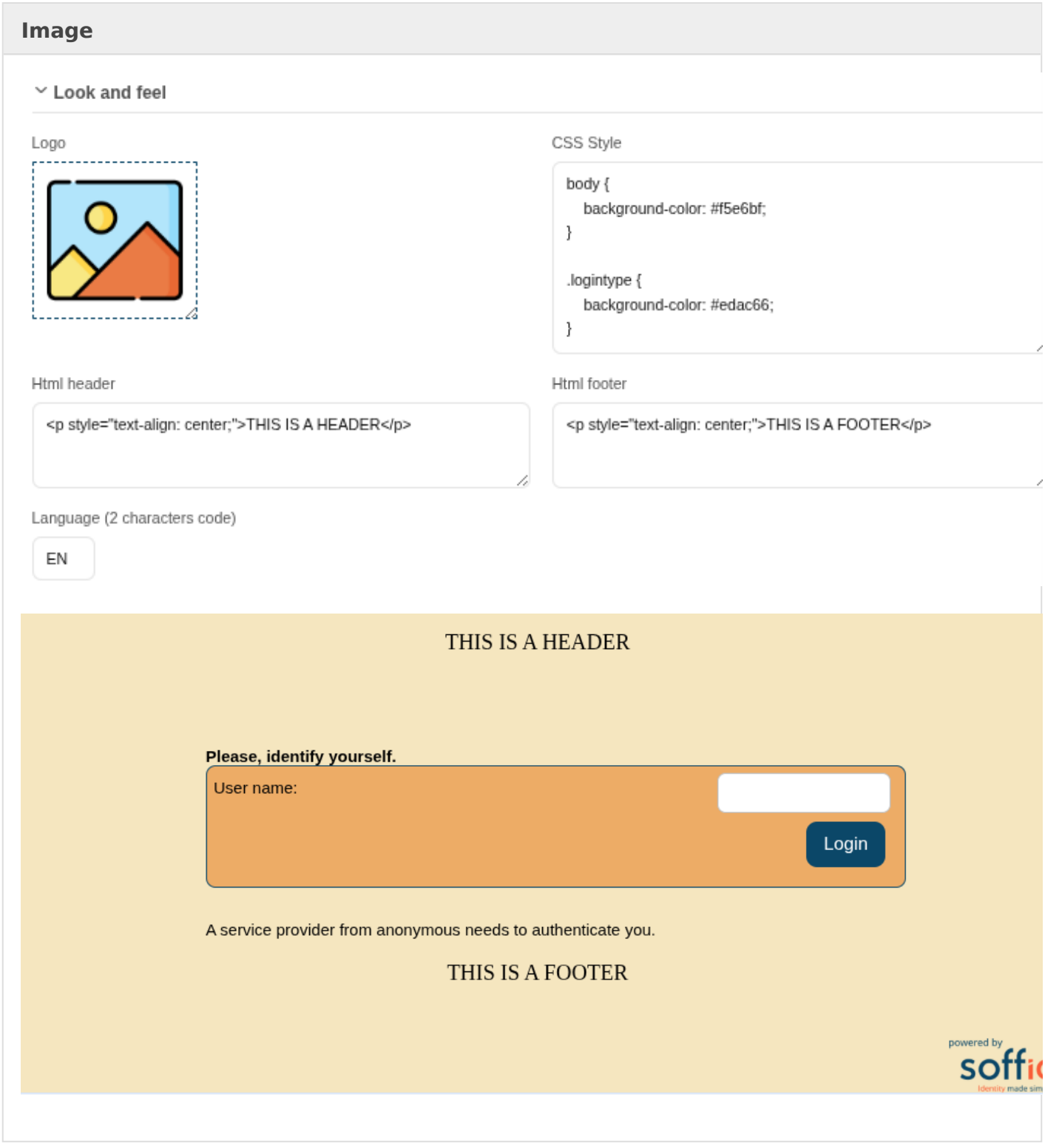
## Look and feel

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 users 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)**: language used by default in the first access

> Restarting the syncserver will be necessary to apply the look and feel changes.




# External SAML IdP

## Identification

- **Idp type**: External SAML IdP (this one has to be selected)
- **Identifier**: 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.
- **Metadata (file)**: from this field, you can directly download a file with the metadata.

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.

## 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.

# OpenID-Connect

## Identification

- **Idp type**: OpenID Connect (this one has to be selected)
- **Identifier**: 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**: 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

- **Idp type**: Facebook (this one has to be selected)
- **Identifier**: 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

- **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

- **Idp type**: Google (this one has to be selected).
- **Identifier**: 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

- **Idp type**: Linkedin (this one has to be selected)
- **Identifier**: 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.

# Virtual identity provider

## Identification

- **Identifier**: unique name to identify the identity provider.
- **Name**: user friendly name to identify the identity provider.
- **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.
- **Metadata (file)**: from this field, you can directly download a file with the metadata.

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

## SAML Security

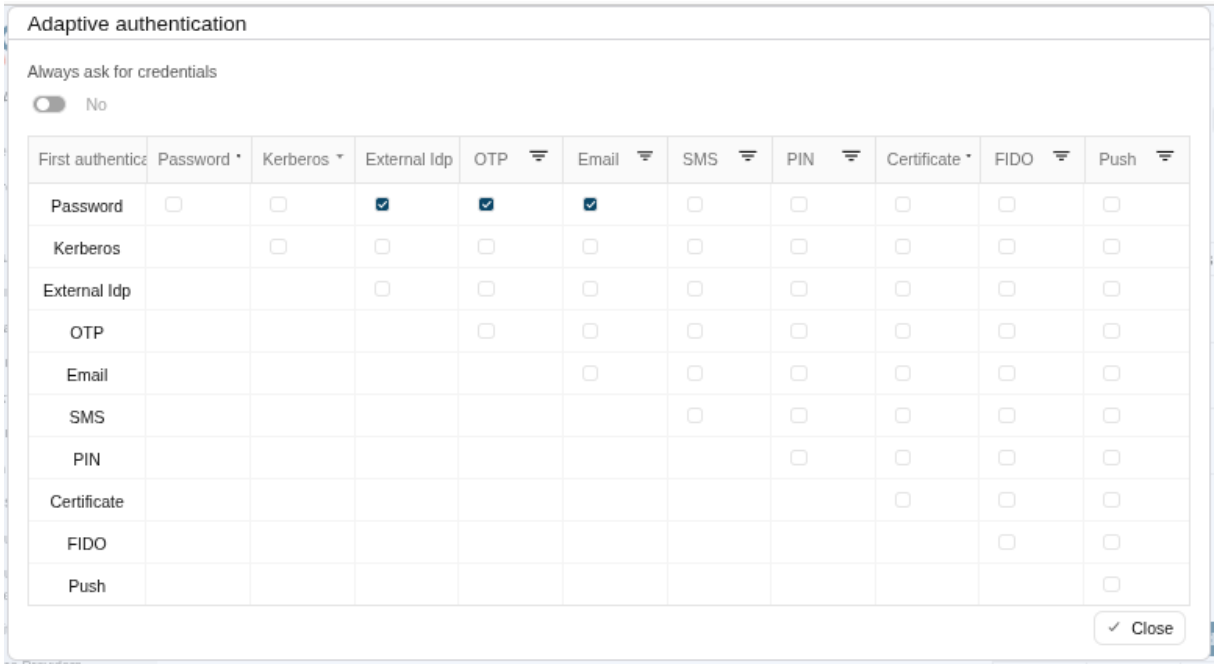
- **Public key**:
  - **Generate public/private key**:
    - **Delete public/private key**: allows you to delete the public/private key generated previously.
    - **Generate PKCS10**: generates a PKCS10 file (Certification request standard)
  - **Upload PKCS12 file**: allows you to upload a PKCS#12 file. That file must contain the private and public kesus and the server certificate as weel. Mind that PKCS#12 file use to be protected by a PIN.
- **Certificate chain**: text certificate chain created with one of the previous options.

### Authentication

- **Default authentication methods**: the button open a popup.
  - **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.
  - **"Matrix of 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

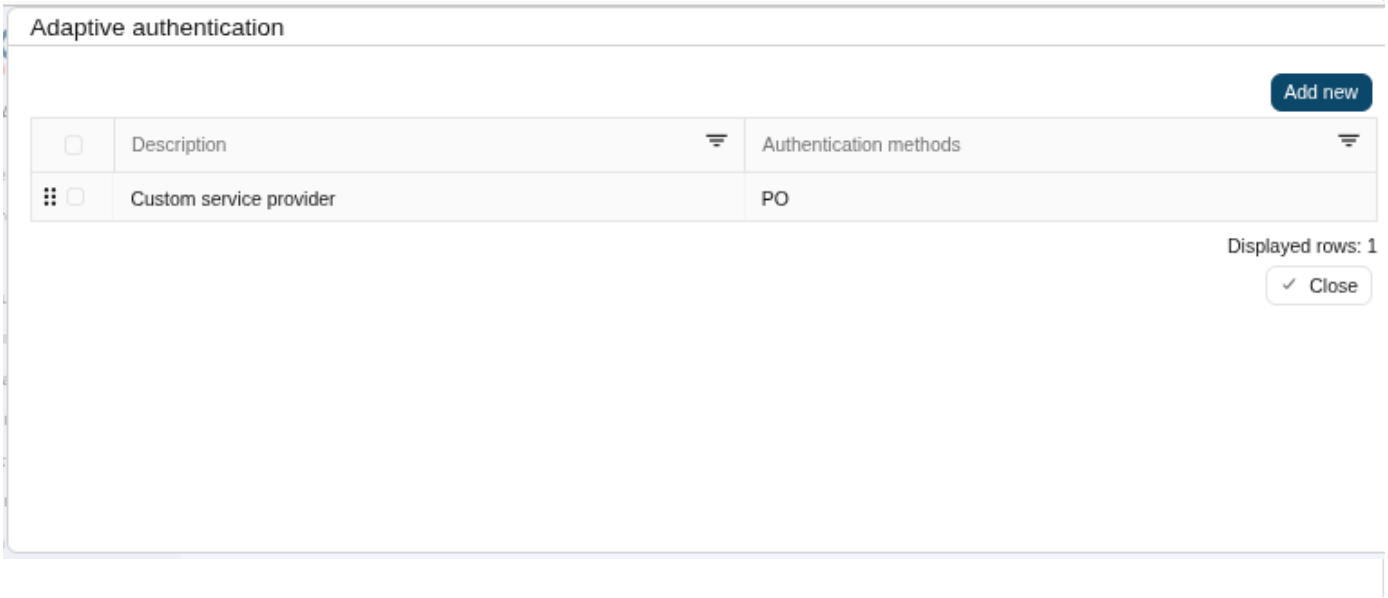
**Image**

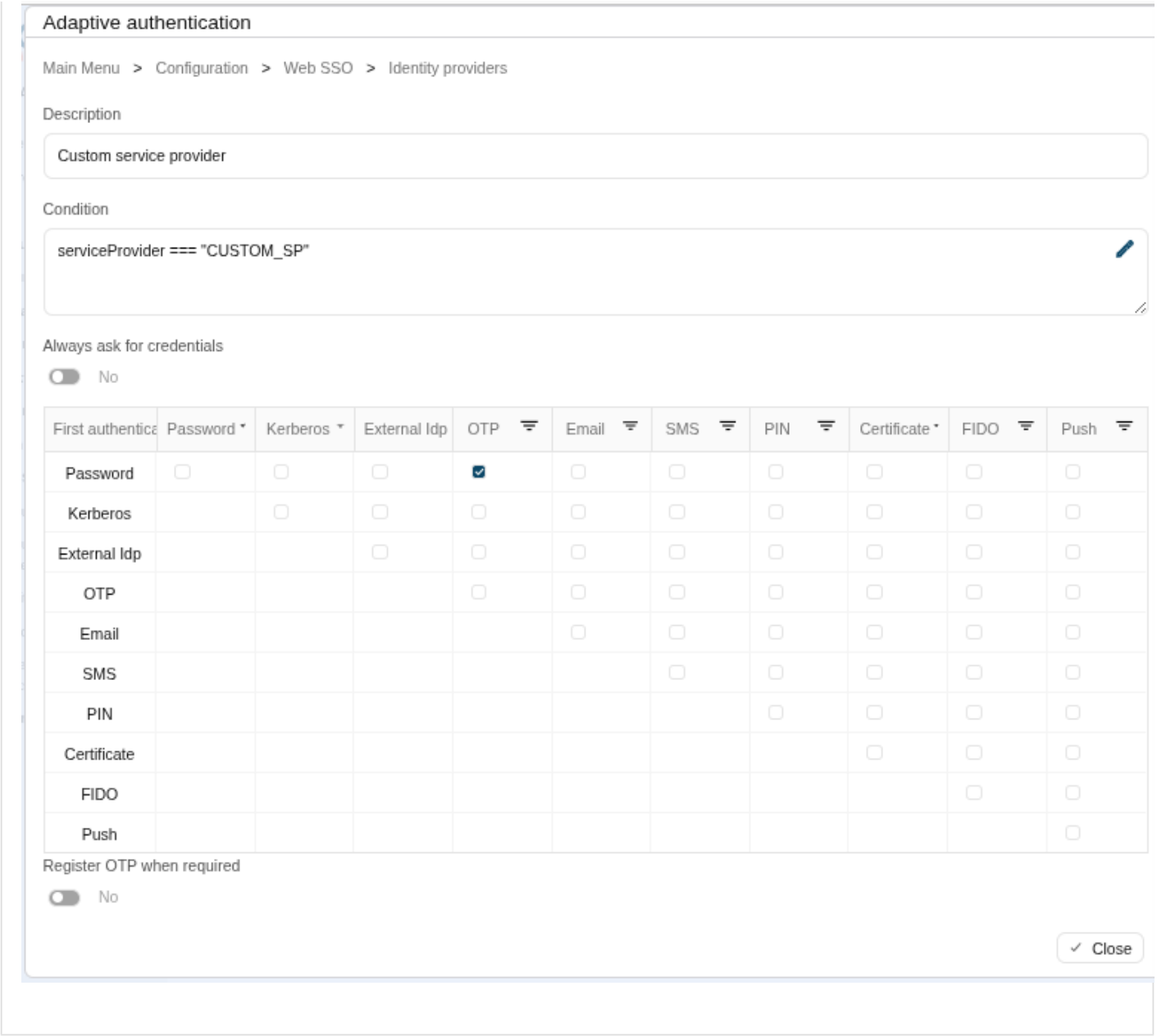



- **Adaptive authentication**: the button open a popup.
  - **"Table of adaptive authentication"**
    - **Description**: description of the adaptive authentication.
    - **Authentication methods**: displays the authentication methods seleccted.
  - **"Adaptive authentication popup"**: that option allows 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.
    - **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.

- **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.

**Image**






- **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**: (displayed when Allow users to self-service enabled) identifies the password policy that is to be applied. More information on this link User Type.
- **Primary Group**: (displayed when Allow users to self-service enabled)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 implemented by the Identity Provider. There are some accepted protocols, those allows a custom config dependent on the selected profile

- OpenIDProfile
- SAML1ArtifactResolutionProfile
- SAML1AttributeQueryProfile
- SAML2ArtifactResolutionProfile
- SAML2AttributeQueryProfile
- SAML2ECPProfile
- SAML2SSOProfile

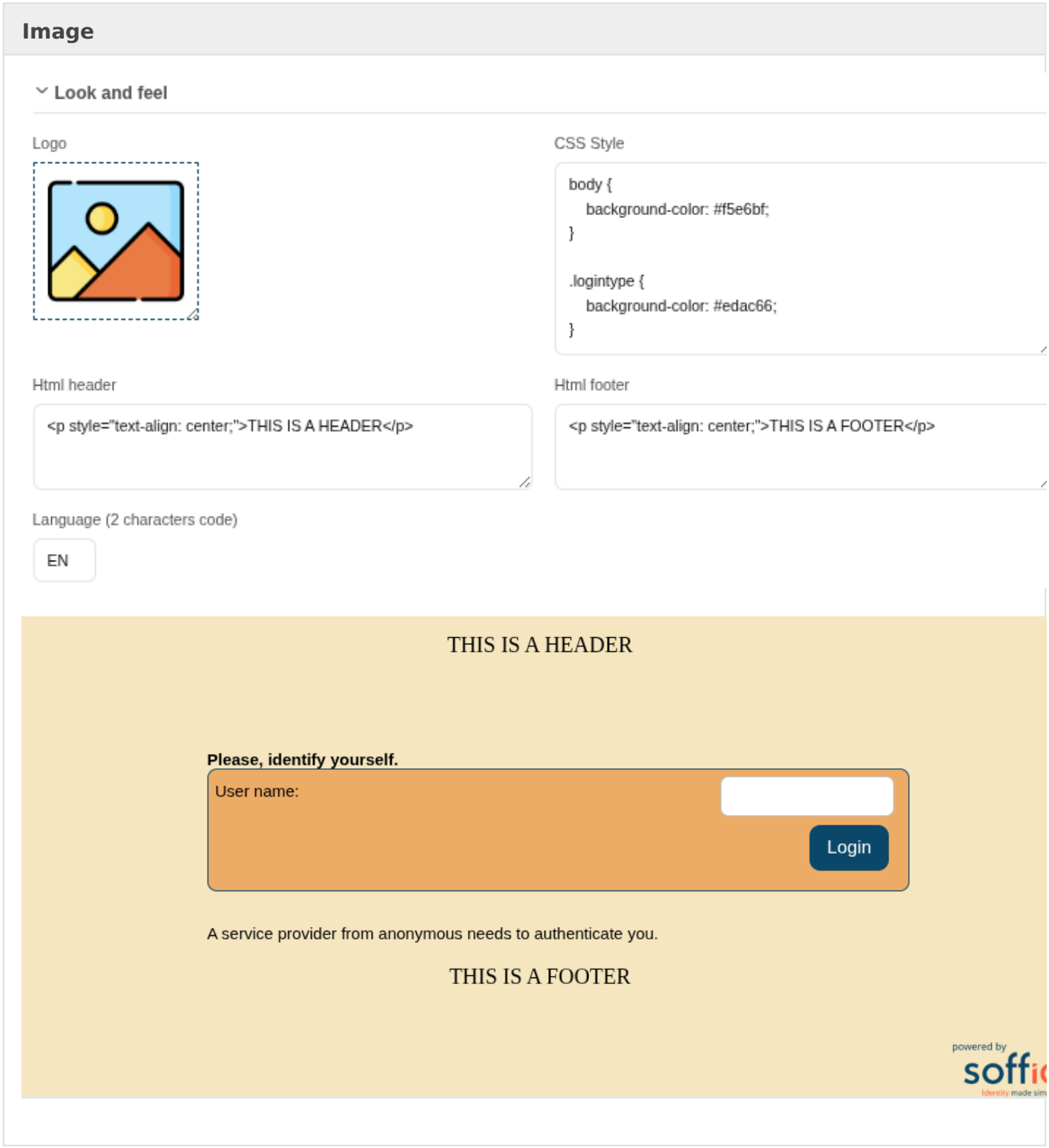
> You can visit the Profiles chapter for more information about each one.

## 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 users 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)**: language used by default in the first access

Restarting the syncserver will be necessary to apply the look and feel changes.




## Service Providers

It will be necessary to bind any service provider to the virtual identity provider. When no such bind exists for a service provider, the actual identity provider profile configuration applies.

- **Name**: name of the service provider

# Actions

## Federation tree

| | |
|---|---|
| **Add group** | Allows you to create a new entity group. You can choose that option by clicking on the "Add group" button in the tree, then Soffid will display a new window with the fields to fullfil. To add a new entity group it will be mandatory to fill in the required fields and save or apply changes. |
| **Add identity provider** | Allows you to add a new identity Provider. You must click the "Add identity provider" button, under the proper entity group, then Soffid will display a new window with the data to fulfill for the new identity provider. To add a new identity provider it will be mandatory to fill in the required fields and save or apply changes. |
| **Add virtual identity provider** | Allows you to add a virtual identity provider. You must click the "Add virtual identity provider" button, under the proper identity provider, which has to be a Soffid IdP, then Soffid will display a new window with the data to fulfill for the new virtual identity provider. To add a new virtual identity provider it will be mandatory to fill in the required fields and save or apply changes. |

## Entity group detail

| | |
|---|---|
| **Apply changes (disk button)** | Allows you to save the data of a new entity group or to update the data of a specific entity group. To save the data it will be mandatory to fill in the required fields. |
| **Delete** | Allows you to remove the entity group. You can find this option in the "three points" menu by clicking on the "Delete" button. To perform that action, Soffid will ask you for confirmation, you could confirm or cancel the operation. |
| **Undo** | Allows you to quit without applying any changes. |
| **Apply changes** | Allows you to save the data of a new entity group or to update the data of a specific entity group. Once you apply changes, the plugin details page will be closed. |

## Identity provider detail

| | |
|---|---|
| **Save** | Allows you to save the data of a new identity provider or to update the data of a specific identity provider. To save the data it will be mandatory to fill in the required fields. |
| **Delete identity provider** | Allows you to delete the identity provider. To delete an identity provider you can click on the "three points" icon and then click the delete button. Soffid will ask you for confirmation to perform that action, you could confirm or cancel the operation. |
| **Undo** | Allows you to quit without applying any changes made. |
| **Apply changes** | Allows you to save the data of a new identity provider or to update the data of a specific identity provider and quit. To save the data it will be mandatory to fill in the required fields. |

# Virtual identity provider detail

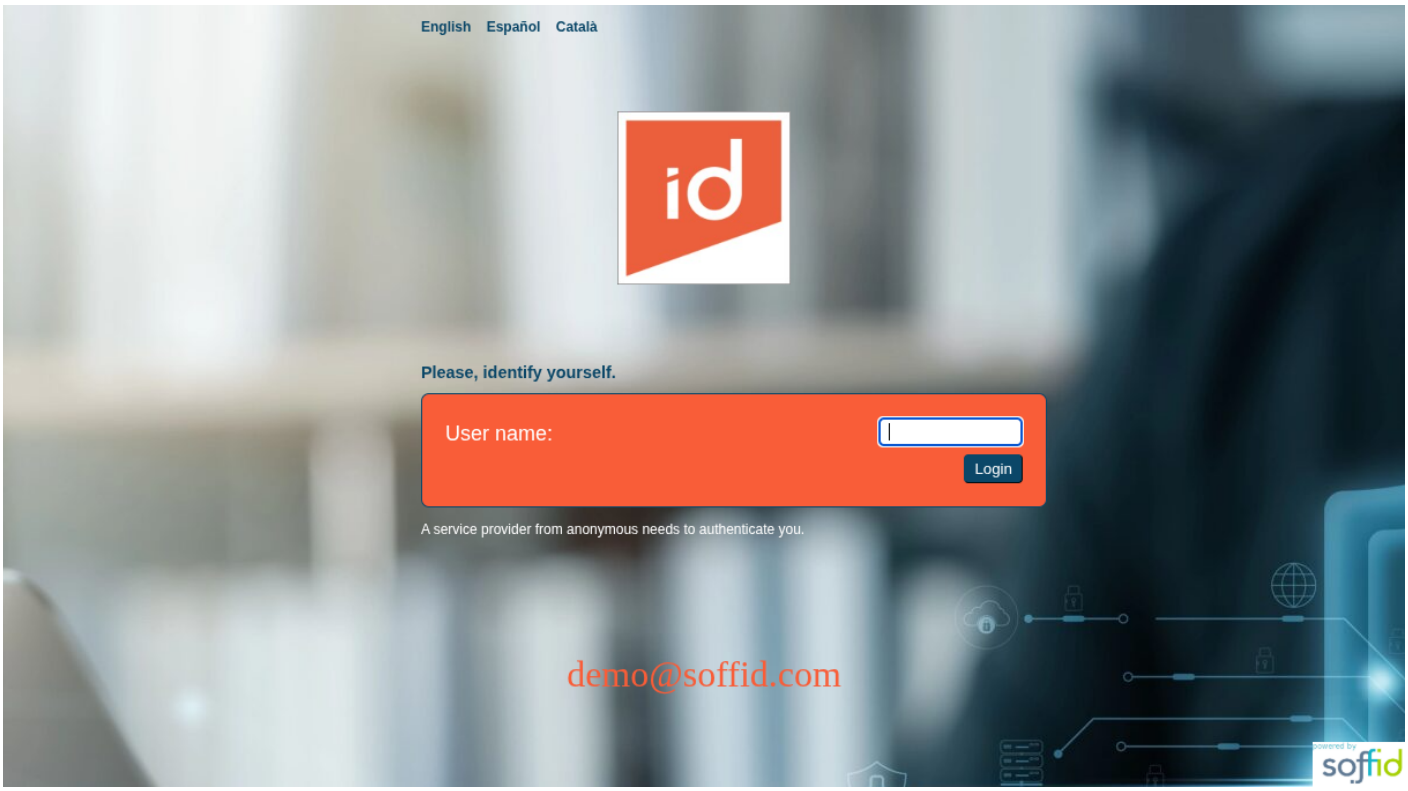
| | |
|---|---|
| **Save** | Allows you to save the data of a new virtual identity provider or to update the data of a specific virtual identity provider. To save the data it will be mandatory to fill in the required fields. |
| **Delete identity provider** | Allows you to delete the virtual identity provider. To delete a virtual identity provider you can click on the "three points" icon and then click the delete button. Soffid will ask you for confirmation to perform that action, you could confirm or cancel the operation. |
| **Undo** | Allows you to quit without applying any changes made. |
| **Apply changes** | Allows you to save the data of a new virtual identity provider or to update the data of a specific virtual identity provider and quit. To save the data it will be mandatory to fill in the required fields. |

# Examples

## Look and feel customisation

In this example, we are going to use all styles except the header, so we can take advantage of the language change and use the manually uploaded logo.

This is the result.




This is the configuration.


CSS Style:

```
body {
  color: white;
  background-image: url("https://www.soffid.com/wp-content/uploads/2025/05/Depositphotos_795124038_XL-1-scaled.jpg");
}

#language a {
  text-decoration: none;
  font-weight: bold;
  color: #0B4768;
```

```
}

p.biglogo img{
  margin-top: 50px;
  width: 150px;
}

p.header {
  color: #0B4768;
  padding-bottom: 10px;
  font-size: larger;
}

.logintype {
  background-color: #F95D38;
  border: 1px solid #0B4768;
  color: white;
  font-size: large;
  padding: 20px;
}

.nologintype {
  color: #0B4768;
  font-size: large;
  padding: 20px;
}

input {
  padding: 4px 8px 4px 8px;
  border-radius: 4px;
  border-color: #0B4768;
  border-width: 1px;
  cursor: pointer;
}

input[type=submit] {
  background-color: #0B4768;
  color: white;
}
```

Html footer:

```
<p style="text-align:center;color: #F95D38;font-size: xx-large;margin-top:100px;">demo@soffid.com</p>
```

If you use the header, the language change options disappear and the logo is not displayed either. You can add the logo yourself using HTML/CSS.

```
<div style="text-align: center;margin-top: 50px;">
  <img src="https://media.licdn.com/dms/image/v2/D4D0BAQEQlaVONhPqHw/company-logo_200_200/B4DZeJJh1kH4AI-/0/1750352666329/soffid_logo?e=2147483647&v=beta&t=yCxIGdOteGHza9p2s1jLNogbO0YKpDS-bHzzHMuQwok" style="display: block; margin: 0 auto; width: 150px;">
</div>
```

---

Revision #19
Created 19 July 2025 12:20:58 by Sion Vives
Updated 5 February 2026 18:36:20 by Sion Vives