Configuration and Settings

This topic describes how to configure the system as a SAML service provider. When the system is a SAML service provider, it relies on the SAML identity provider authentication and attribute assertions when users attempt to sign in to ConcreteCMS . We need to configure both the service provider and Identity provider appropriately to talk to each other. To enable SAML single sign-on, you must provide: 1) Identity provider with certain data from our Service provider . 2) Our Service provider with certain data from Identity provider

- Identity provider configuration
- Our Service provider configuration
- Settings and Customization
- Use Custom Certificate

Identity provider configuration

The SAML standard means that a wide range of identity providers will work with our ConcreteCMS SAML Service provider .

In Identity provider (IdP) side, configuration instructions will vary depending on the vendor, as an administrator please refer to the Identity provider vendor-specific documentation for details. It may have relevant documentation and it may be generic SAML documentation, or specifically targeted for specific Service provider.

List of some of the Identity Providers with links refer to its official documentation to configure a SAML integration.:

- Auth0
- ADFS (Active Directory Federation Services)
- OneLogin
- Okta
- Salesforce
-

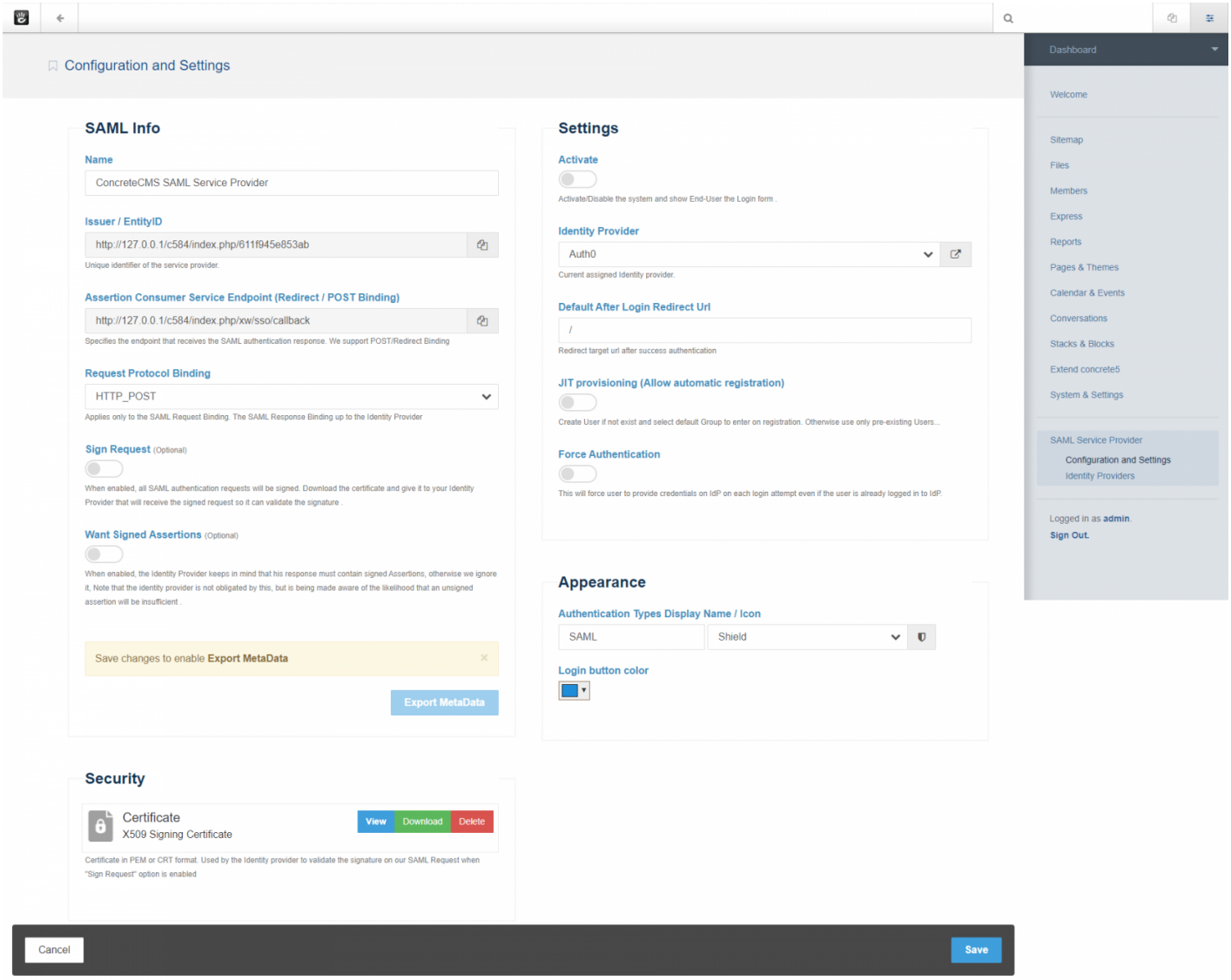
SecureAuth

- Centrify
- simpleSAMLphp

When configuring your identity provider, please consider the notes below to help avoid common issues and as a guide for terminology used .

Our ConcreteCMS SAML Service provider needs to provide some informations to the Identity Provider .

Go to " *Dashboard > SAML Service Provider > Configuration and Setting*s " page .



Add the SAML informations on this page to the Identity Provider (IdP) administration side page so the tenant knows how to receive and respond to our SAML authentication requests.

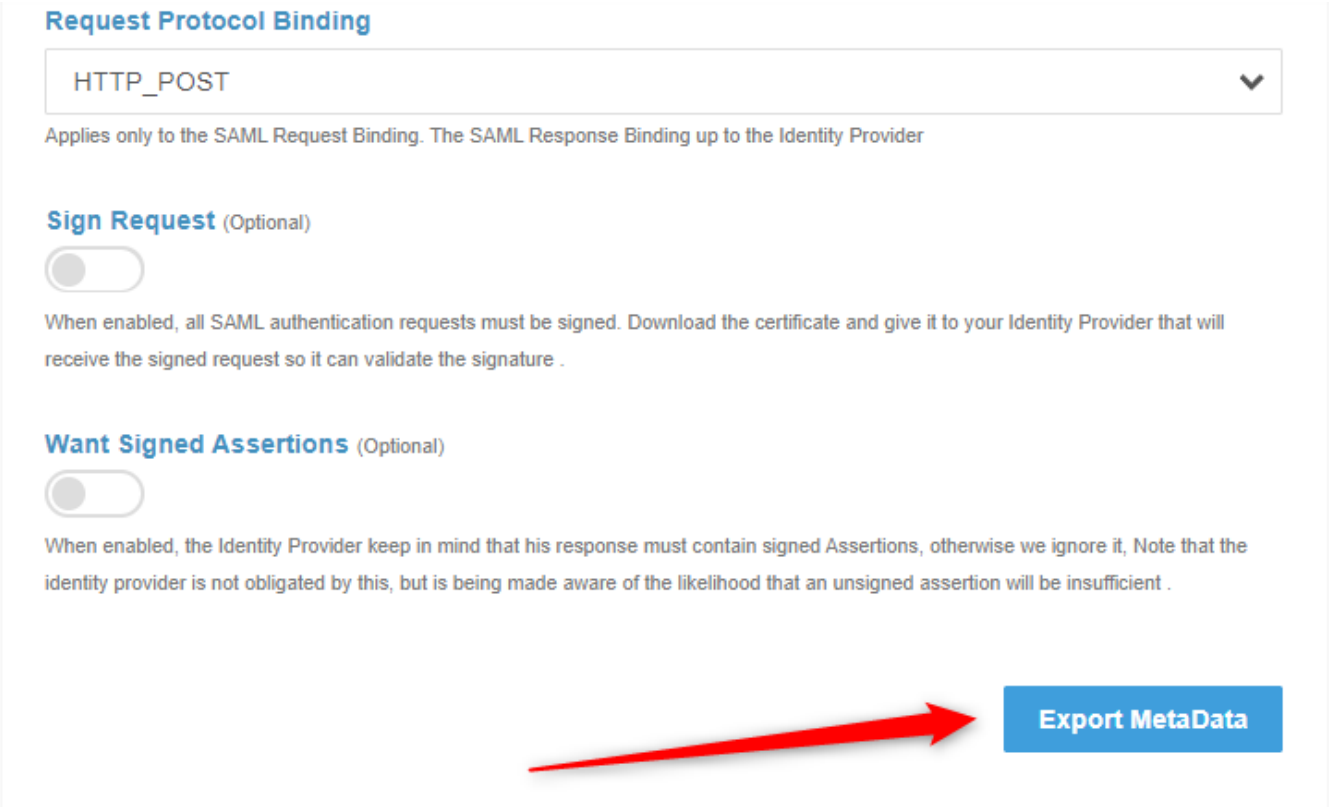
If the IdP supports uploading a Metadata file, you can simply provide the file obtained in the step below.

> SAML Metadata file is the standard format for exchanging configuration information between SAML service provider and the Identity provider . SAML metadata is supplied to partner Identity providers so they can update their configuration .

Click ***Export Metadata*** button in the bottom of ***SAML Info*** section to download an XML file of your SAML configuration settings to send to

your Identity provider .

> You must **Save** new changes to enable **Export** again If you already change some values .



The Identity provider can then upload these configuration settings to connect to our SAML Service provider package .

If the IdP does not support uploading a Metadata file, you can configure it manually as follows. You will need to use some of this informations from this screen to configure it .

SAML Offset Minutes	*Set to make up for time differences between devices.* *This value is prepopulated. It is generated by the system : 5 minutes.*

Our service provider (SP) requires certain attribute information to be received from the IDP when a user signs in using SAML logins. The **NameID** attribute is mandatory and must be sent by your IDP in the SAML response to make the federation with ConcreteCMS work. Since Our SP uses the value of **NameID** to uniquely identify a named

user, it is recommended that you use a **Email** format value.

The IdP needs to pass certain information in order for our SP to either create an account, or match the login information to an existing account. **Email** is the minimum amount of information that needs to be passed. If the IdP is not providing this information, all SAML requests fail. Make sure this information is provided.

We automatically uses the SAML **NameID** to identify users in ConcreteCMS . We recommend setting and configure your IdP so that **NameID** format is **Email Address** .

We specifies `urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress` for the format of the **NameIDPolicy** in assertion requests.

At a minimum, If there is no **NameID** element with **Email Address** Format, the user's email address must be specified as an Assertion Attribute. The name of the Attribute that specifies user email address must be configured as **email** or **mail** or **http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress**

Contact the administrator of the identity provider if you need help determining which source of metadata information you need to provide.

No matter what **Request Binding** you select, the SAML response up to Identity provider side configuration . We currently support **POST** and **Redirect** Binding fror SAML response .

Most **Identity providers** requires at least the following : **Assertion Consume Service (ASC) Endpoint .**

Requests and responses must conform to the SAML protocols for exchanging information .

Your IdP must support SAML 2.0 to connect with our ConcreteCMS SAML service provider .

Some I*dentity providers* cannot accept a signed *authentication request (when* ***Sign Request*** *option is enabled) .*

Sign Request *is optional. Some Idps does not validate signed authentication requests even a signature is present.*

If the **Want Assertion Signed** flag is set and neither the SAML response nor SAML assertion is
signed or the signature cannot be verified, this is considered an error .

However, certain changes in the Service provider will impact your SAML configuration. If any of these changes occur, the metadata is automatically updated on your SP side, but you will need to update the information on the Identity Provider side so that message exchange can occur successfully.

As always with **SAML2**, you can't expect all **Idps** to support everything. you have to test if your Idp supports some required options

Many SAML terms can vary between providers. It is possible that the information you are looking for is listed under another name. For more information, start with your identity provider’s documentation. Look for their options and examples to see how they configure SAML. This can provide hints on what you’ll need to configure our Service provider to work with these providers.

The following articles outline configuration instructions for four common third-party Identity providers:

- Configuring **Auth0** SAML Identity provider
- Configuring **OneLogin** SAML Identity provider
- Configuring **Keyloak** SAML Identity provider

Our Service provider configuration

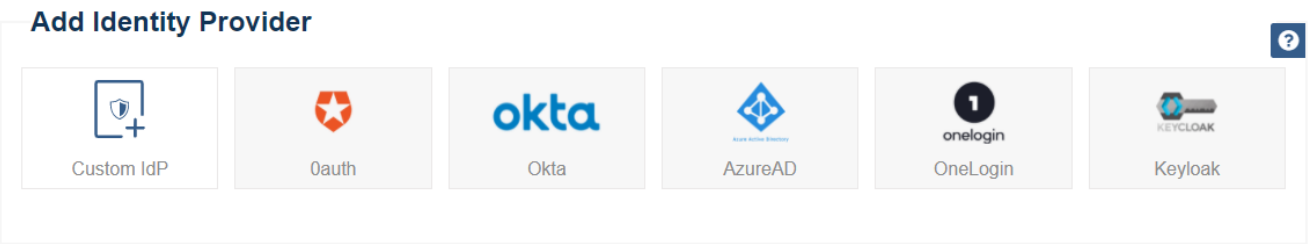
After doing configuration In Identity provider side (previous page), we need to configure our Service provider to complete the SAML setup.

This article provides an generic example walk-through of configuring an Identity Provider (IdP) in the system .

> It is recommended that you or administrators already configuired an **IdP** from his vendor-specific side before proceeding...

Go to " Dashboard > SAML Service Provider > Identity providers " page

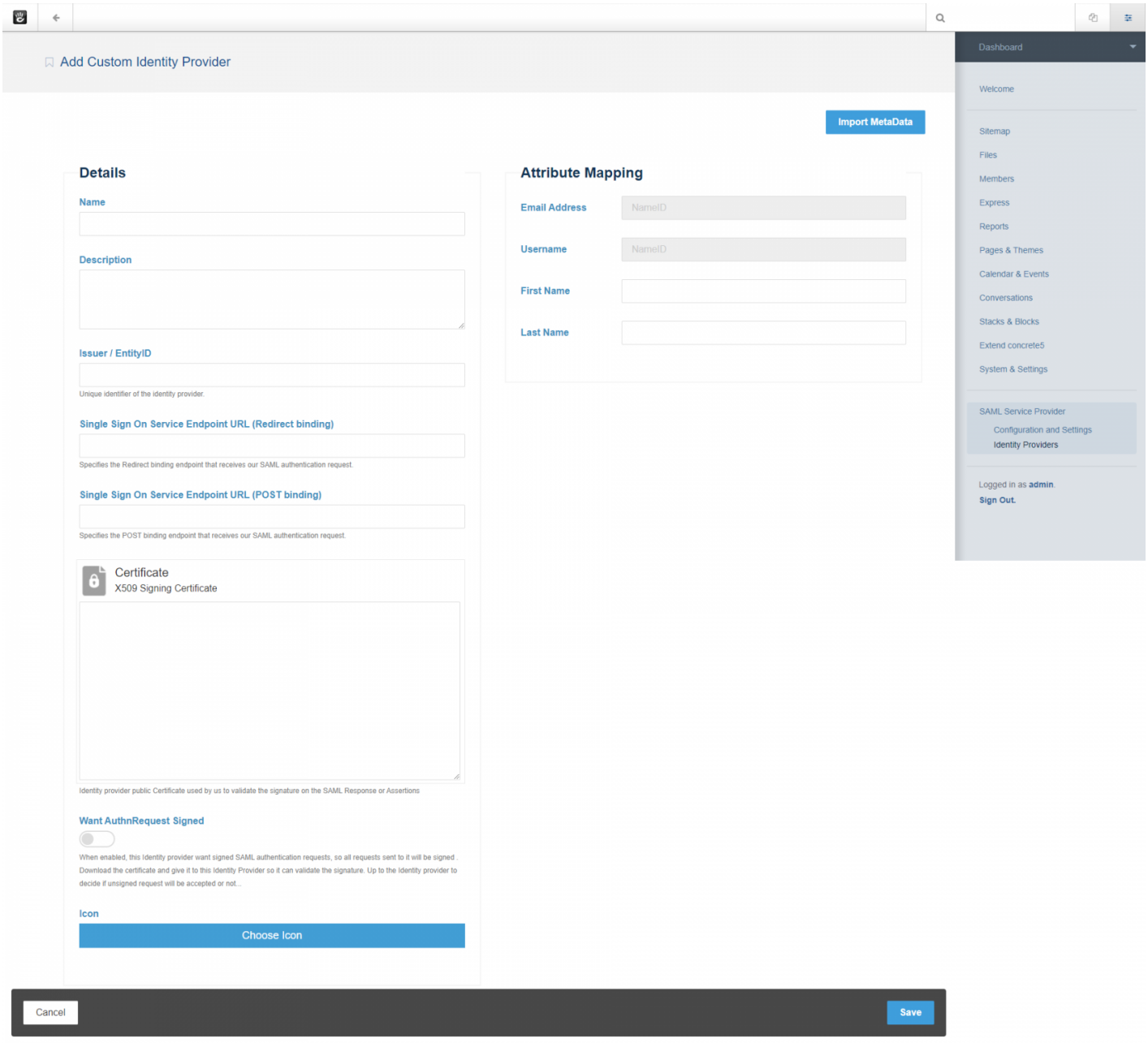
Select your Identity provider, If you don't see your targeted provider listed, choose "*Custom IdP*" .



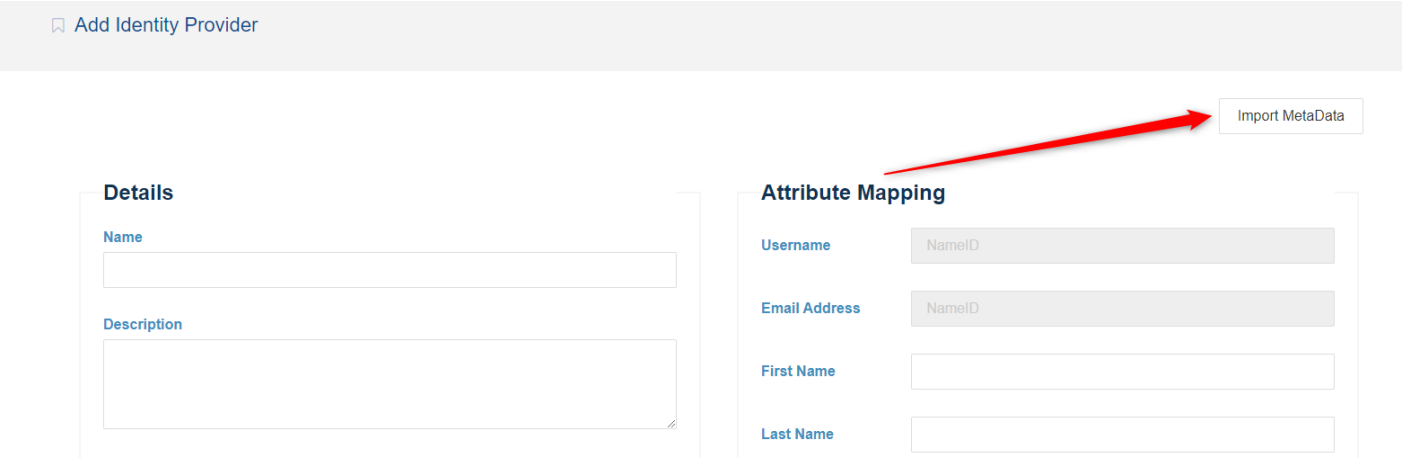
The following values must be provided, and there’s often quite a few

of them. but as an administrator you'll need to provide at least some of this informations:

Name, EntityID, Single Sign On Service Endpoint (Redirect binding) or **(POST binding)** and **X509 Certificate**

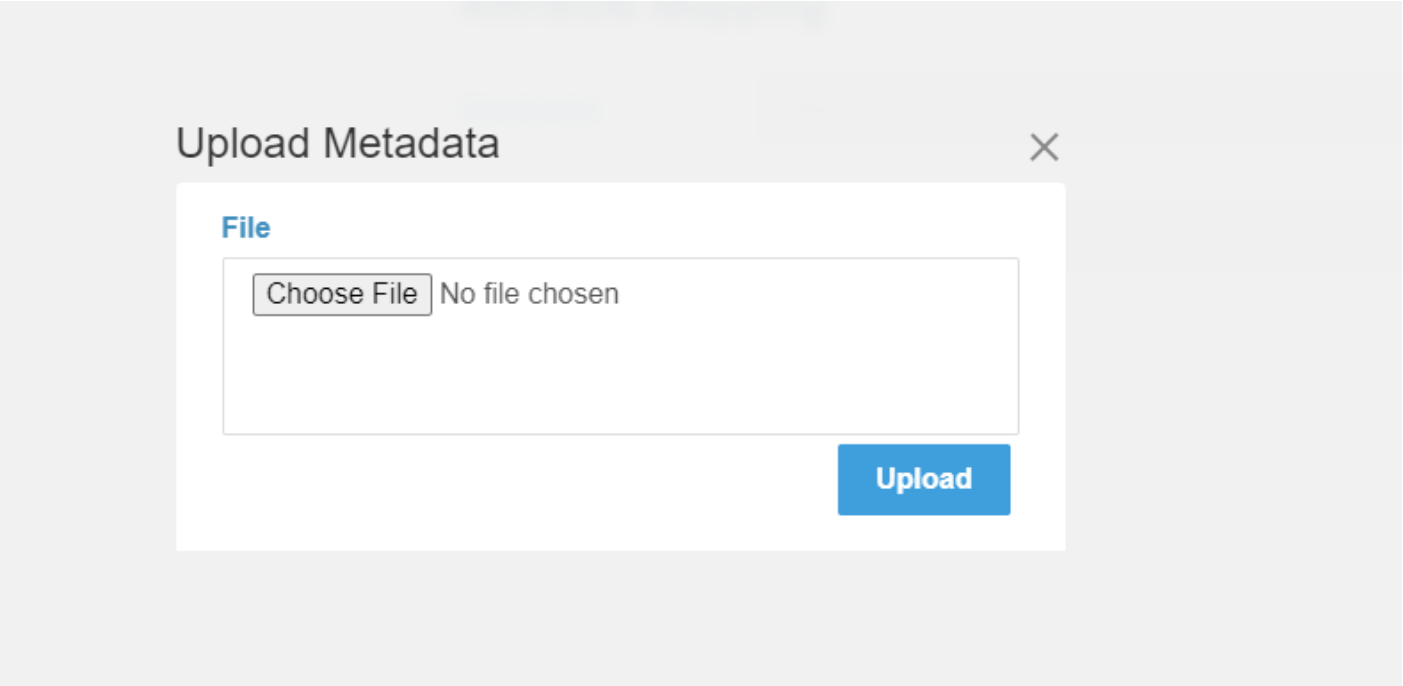


Some Identity providers may offer an Metadata XML document during the configuration process on their sides . This file contains all the information requested in following sections. If you have this file, you can click in ***Import Metadata*** button .



And you can now upload it . Select that file and click in ***Upload*** button,

and the system will parse it to populate the required fields in following sections.



Alternatively, You can fill out the required fields from the output

obtained during your specific Identity provider (IdP) side configuration.

1) Details section

FIELD	**DESCRIPTION**
Issuer / EntityID	*The unique identifier of the Identity p*
Single Sign On Service Endpoints (POST / Redirect)	*URL's where our service provider ser SAML request to start the login seque* *One endpoint URL at least is required (**POST** Or **Rea***
Signing Certificate	*The certificate that Identity provider u digitally signe a SAML response / ass Our service provider uses it to valida signature of the SAML authentication response / assertions .*
Want AuthnRequests Signed (optionel)	*Indicate that if this Identity provider w signed SAML request, so all sent requ it will be signed .*
Name	*The display name of the Identity prov as a reference*
Description (Optionel)	*Short description of the Identity provi*
Icon (Optionel)	Image reference of *the Identity provic*

> **Request Protocol Binding** of your Service provider configuration (See **Configuration and Settings page**) use one of this endpoints according to selected binding .

2) Attributes Mapping section

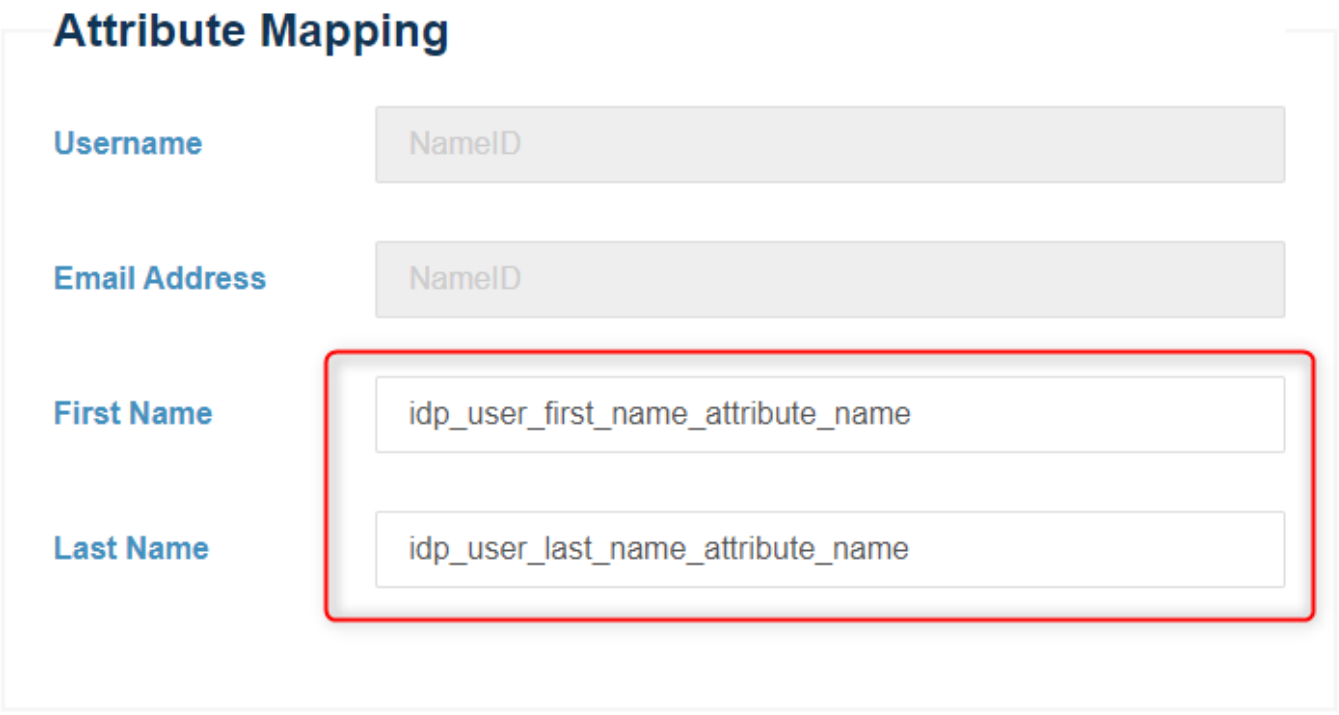
Sometimes the names of the attributes sent by the Identity provider does not
match the names used by user for the ConcreteCMS accounts. In this section we must set the mapping between IdP fields and ConcreteCMS fields .

So this feature allows you to map user attributes sent by the IdP during SSO to the user attributes (first name and last name) at ConcreteCMS .

Fill out mapping fields (First name , Last name) by attributes names obtained during the Identity provider side configuration.

Every attribute must have its own unique representation in a SAML attribute assertion to ensure that there are no misinterpretations or miscommunication. Thus, SAML exchanges rely on consistent attribute naming to deliver information about users in a way that is mutually understood between the IdP and SP. This attribute name must be expected and handled by relying parties.

As a best practice, users should use their emails as the primary connection ID to log on via the SAML plugin because it is always a unique value. While users can be configured to use other attributes such as their first name or last name, these may not always be unique values within an organization.



Configure your IdP so that the **NameID** specifies an element to identify a user. We recommend using **Email** as user Identifier **.**

Note that we automatically use **NameID** value sent by the Identity provider as **Email address** , and to generate **Username**

We fetch users from both **Email Adress** and **Username** and if in both cases the user is not found, a new user is created (If **JIT provisioning option** is enabled)

Note that if **First Name / Last name** inputs are empty we ignore mapping them .

Note that if ConcreteCMS user account doesn't have predefined ConcreteCMS User Attributes with Handle "**first_name**" and "**last_name**" otherwise attributes will not be added and the provided Info ignored . Visit this ConcreteCMS Doc to mannuely add them as an Administrator .

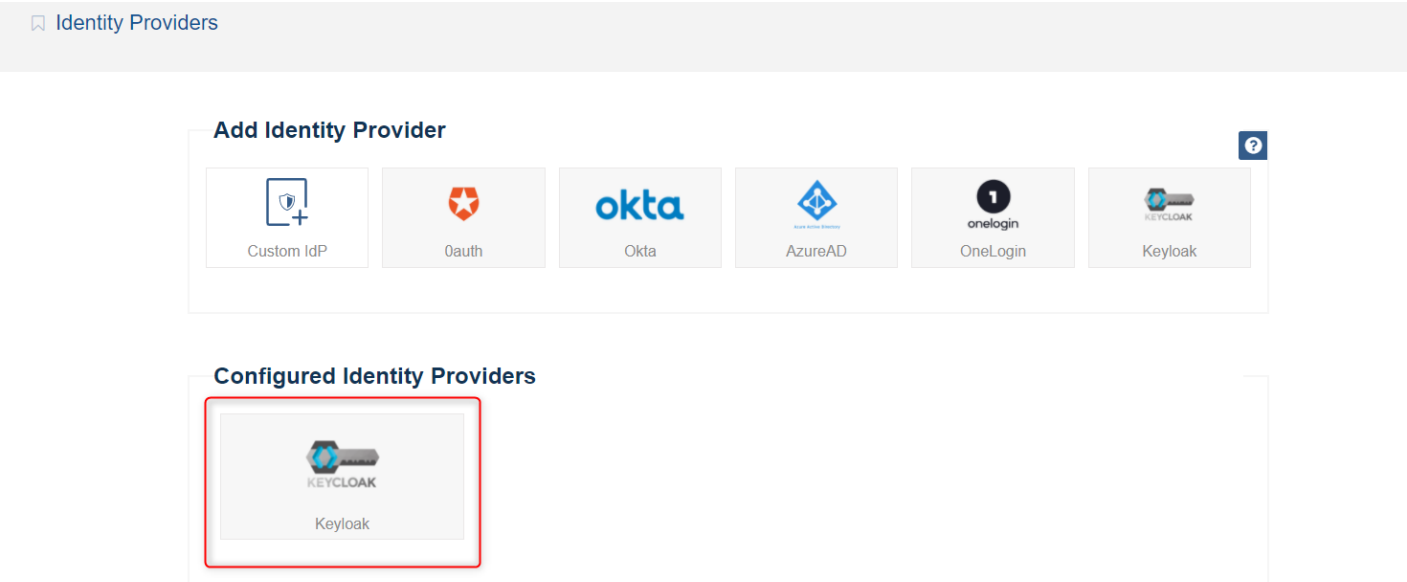
If the configuration is set up correctly . Save your changes by clicking the ***Save*** button in right bottom of your page.

Cancel your changes and go back to main "*Identity providers*" page by clicking the ***Cancel*** button in left bottom of your page.

> And at this point, you have successfully configured an Identity provider in the system.

The new SAML configured IdP is added to your **Configured Identity provider** list

in " Dashboard > SAML Service Provider > Identity providers " page .

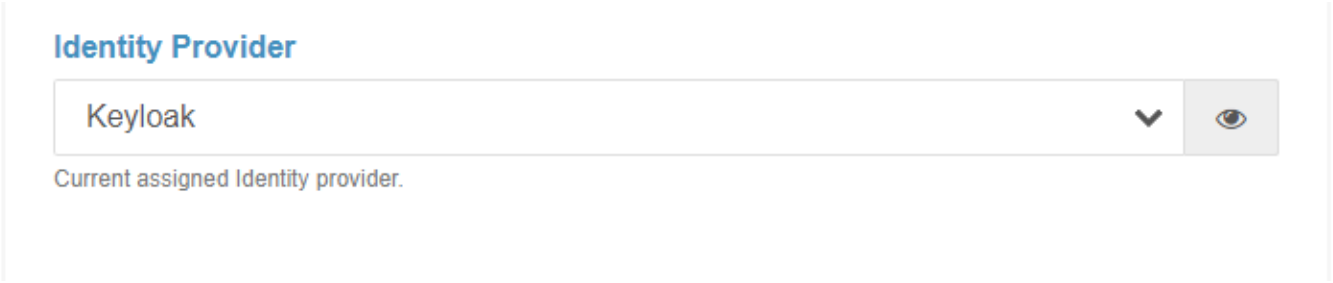


When you've set up a IDP, you can update the settings for it by clicking it .

We need to assign this Identity provider as the active one in the system .

Navigate to " *Dashboard* > *SAML Service Provider* > *Configuration and Setting*s "

Go to the bottom of ***Settings*** section and select your configured identity provider (from step above) appeared in the configured IdPs list .

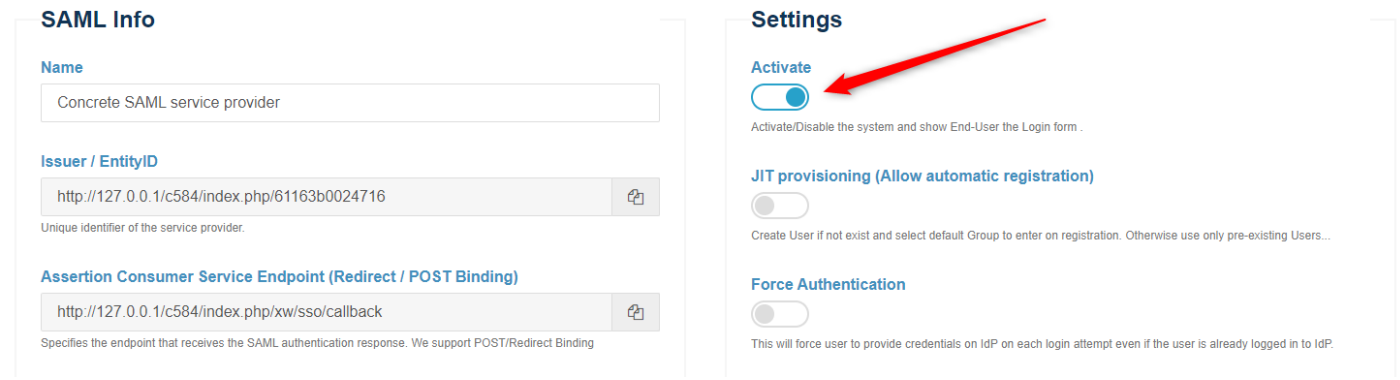


Save your changes by clicking the ***Save*** button in right bottom of your page.

> And at this point, you have successfully activate your Identity provider in the system

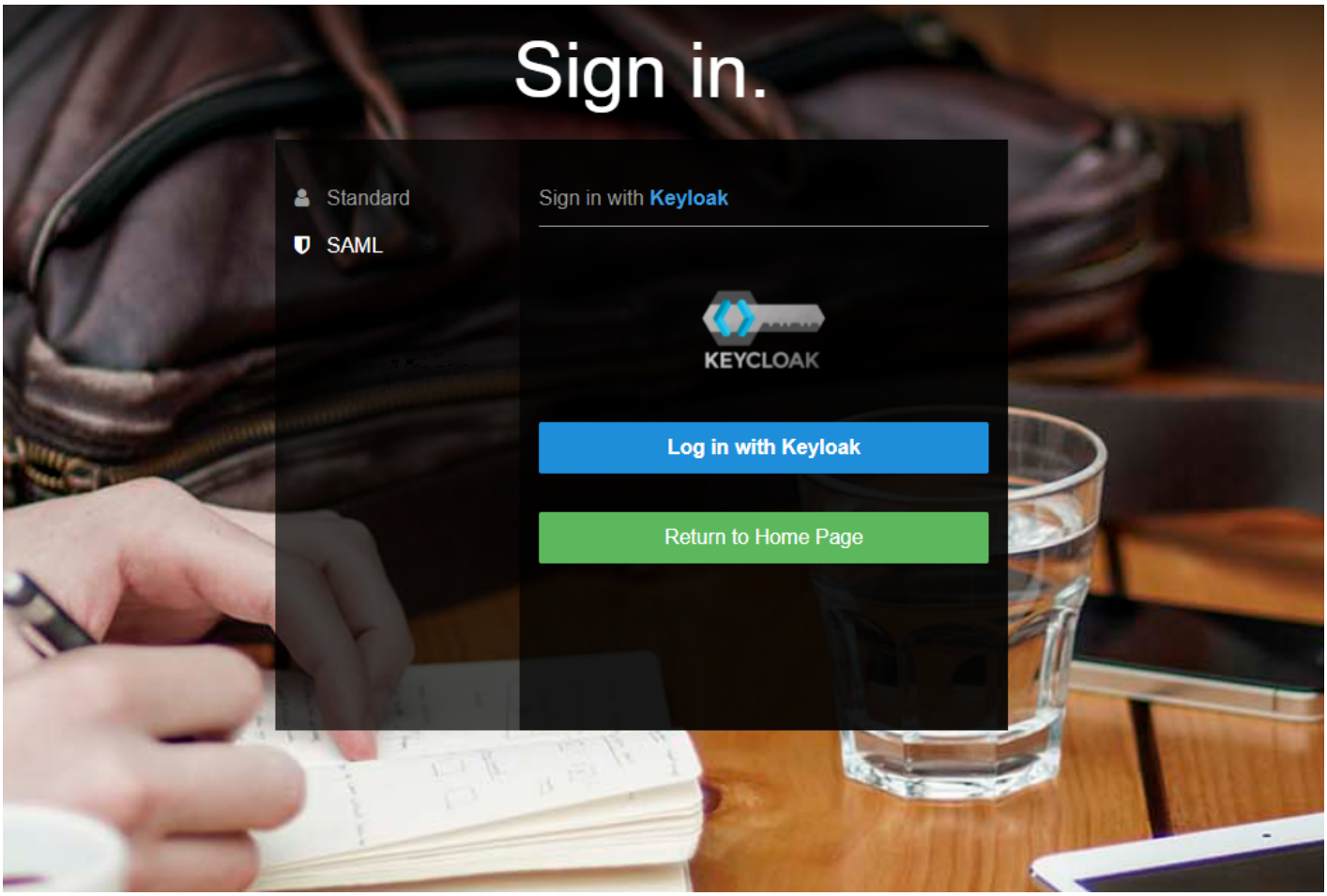
---

Click ***Activate*** *and* ***Save*** to show your End Users the Login form .



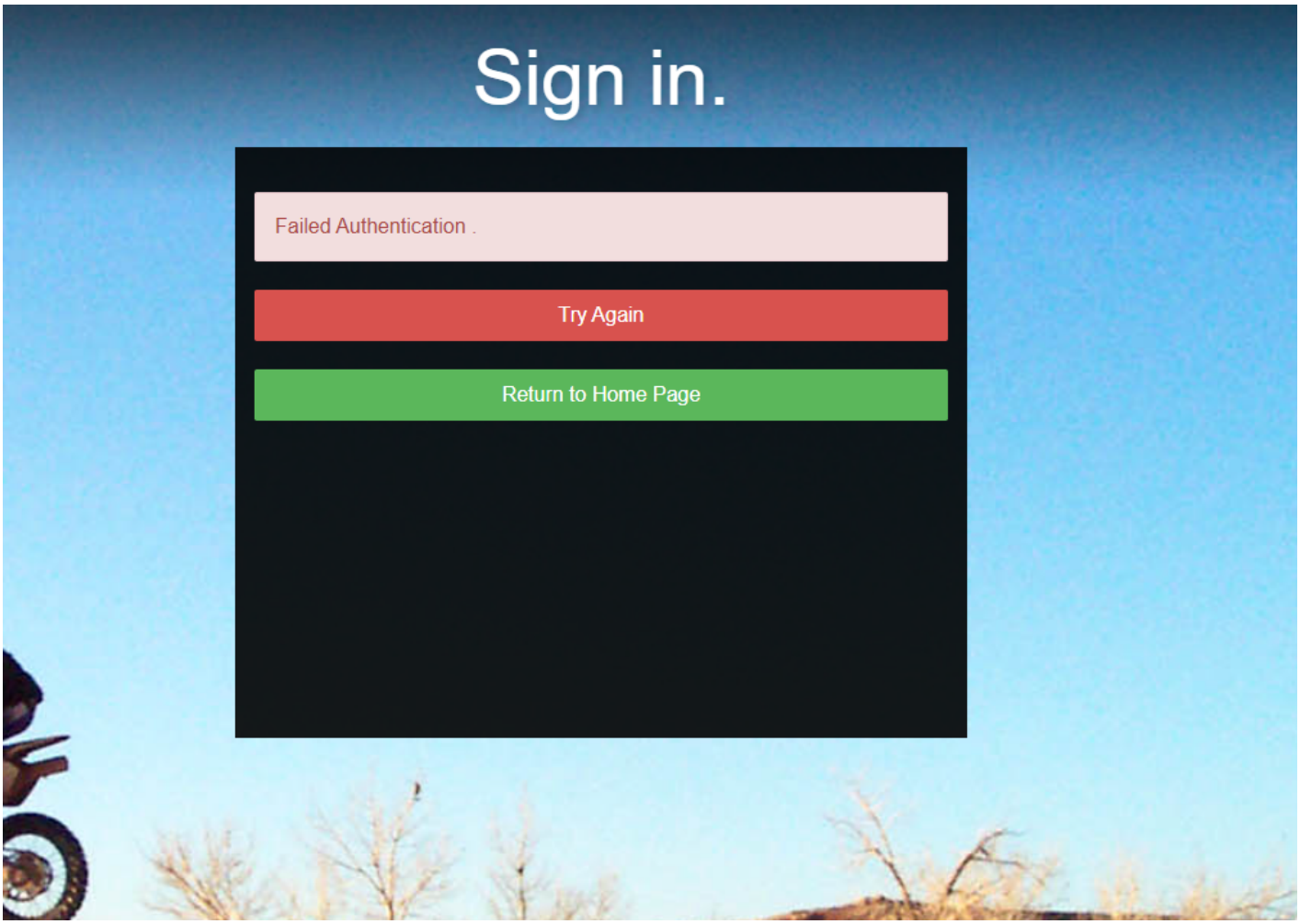
You should now able to see a '**SAML**' option in the ConcreteCMS login screen . This will redirect users to login to the Identity provider instance for their username/password and will create a new ConcreteCMS user account with chosen group (If **JIT provisioning** option is enabled) .

Once you've completed the setup steps, it's important to test to make sure everything is working properly.



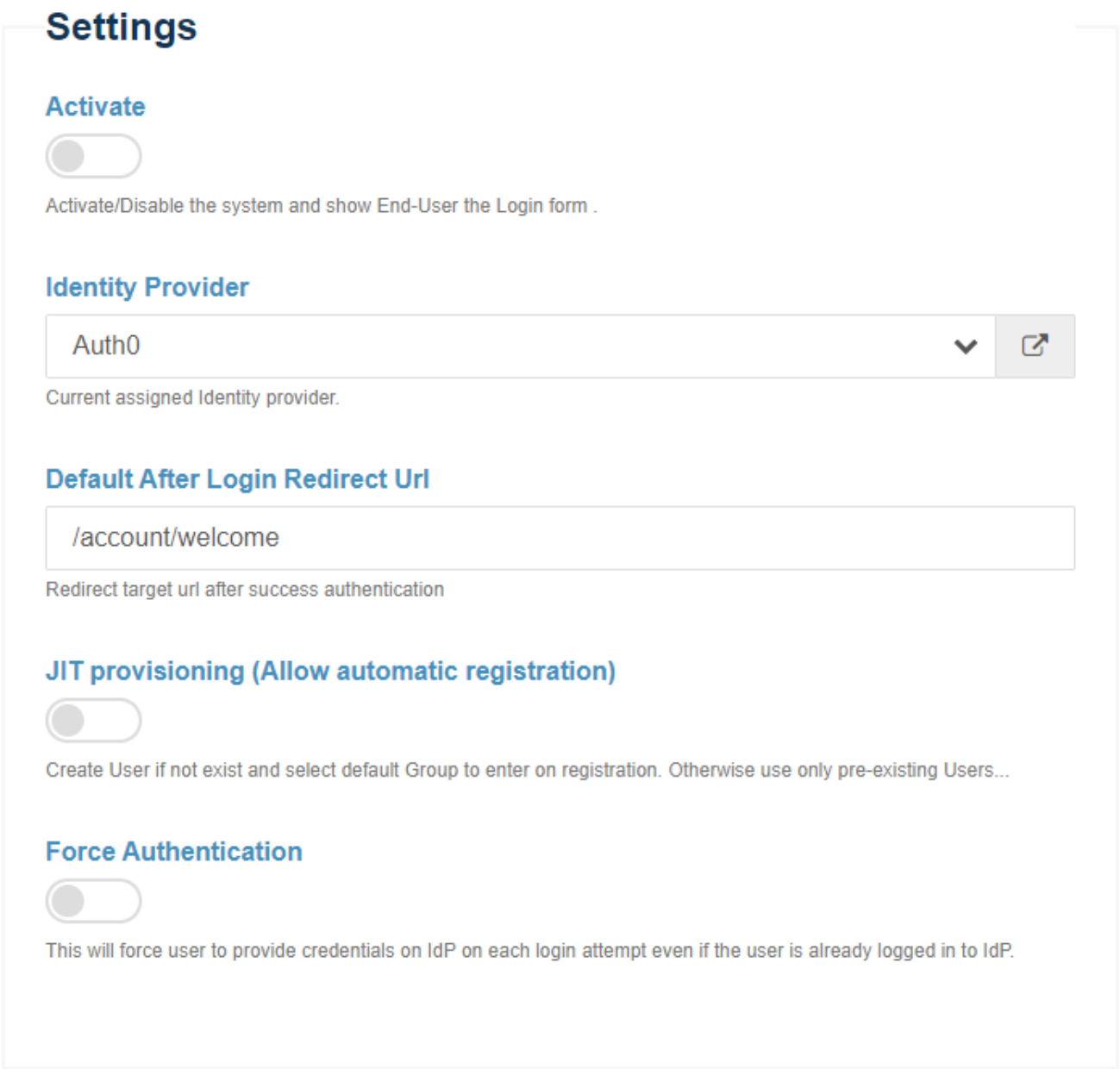
You did it! Your ConcreteCMS Website is configured to provide SAML SSO services. Your users may sign in to your website with the username and password stored by your SAML 2.0 identity provider.

If errors are presented, ensure that all necessary fields have been correctly populated .



Double-check your steps. If you are still having trouble . first check the configuration of your service provider in your side and the identity provider from it vendor side . Also check Troubleshooting && FAQ page to inspect the ConcreteCMS logs.

Settings and Customization



Activate

When enabled, specialized logon page displays when logging on, allowing a user to log on with SAML.

> Easily switch On/Off the SAML Module.

Identity provider

JIT provisioning Option

With JIT provisioning, you can use a SAML Assertion to create users the first time they log in to your concreteCMS from a third-party identity provider. JIT provisioning saves you time and effort because it eliminates the need to provision users or create user accounts in advance. we fetch the user from email (in **NameID** or **Assertion attributes**) and if in both cases the user with that email is not found, a new user is created.

> You can set the default Group for new JIT users. The default role is **None**, but you can choose to add new JIT users as **Administrators** or other new ConcreteCMS created group.

Force authentication

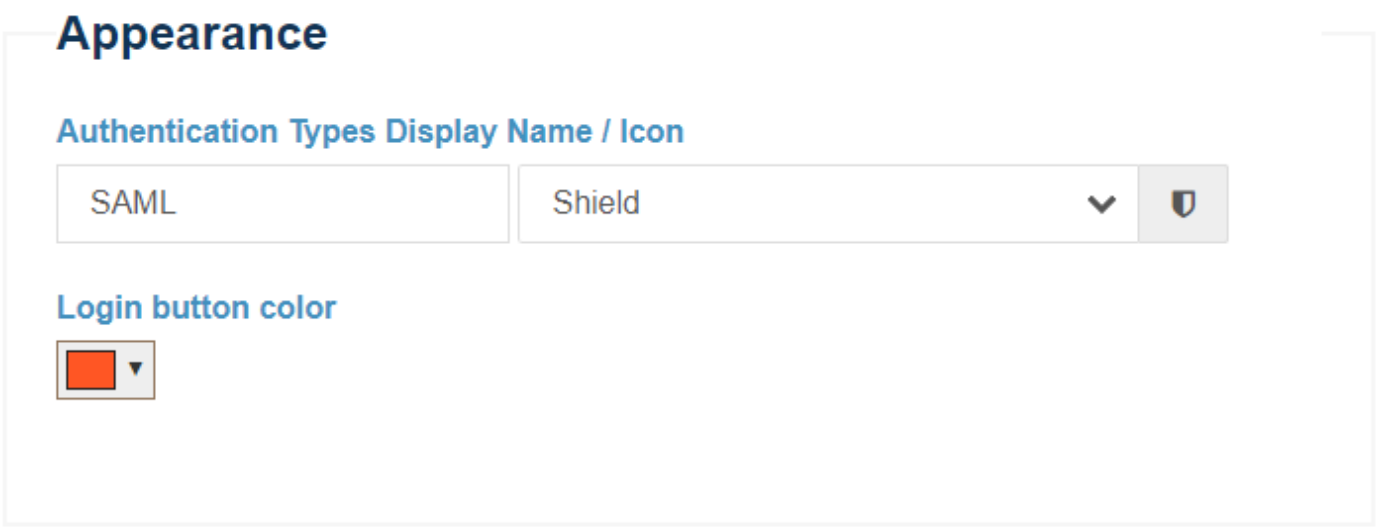
If enabled, users having a current, active SSO session will be re-authenticated by the identity provider .

> Sets the **ForceAuthn** attribute on generated SAML requests, requesting that the IdP re-authenticate the user.

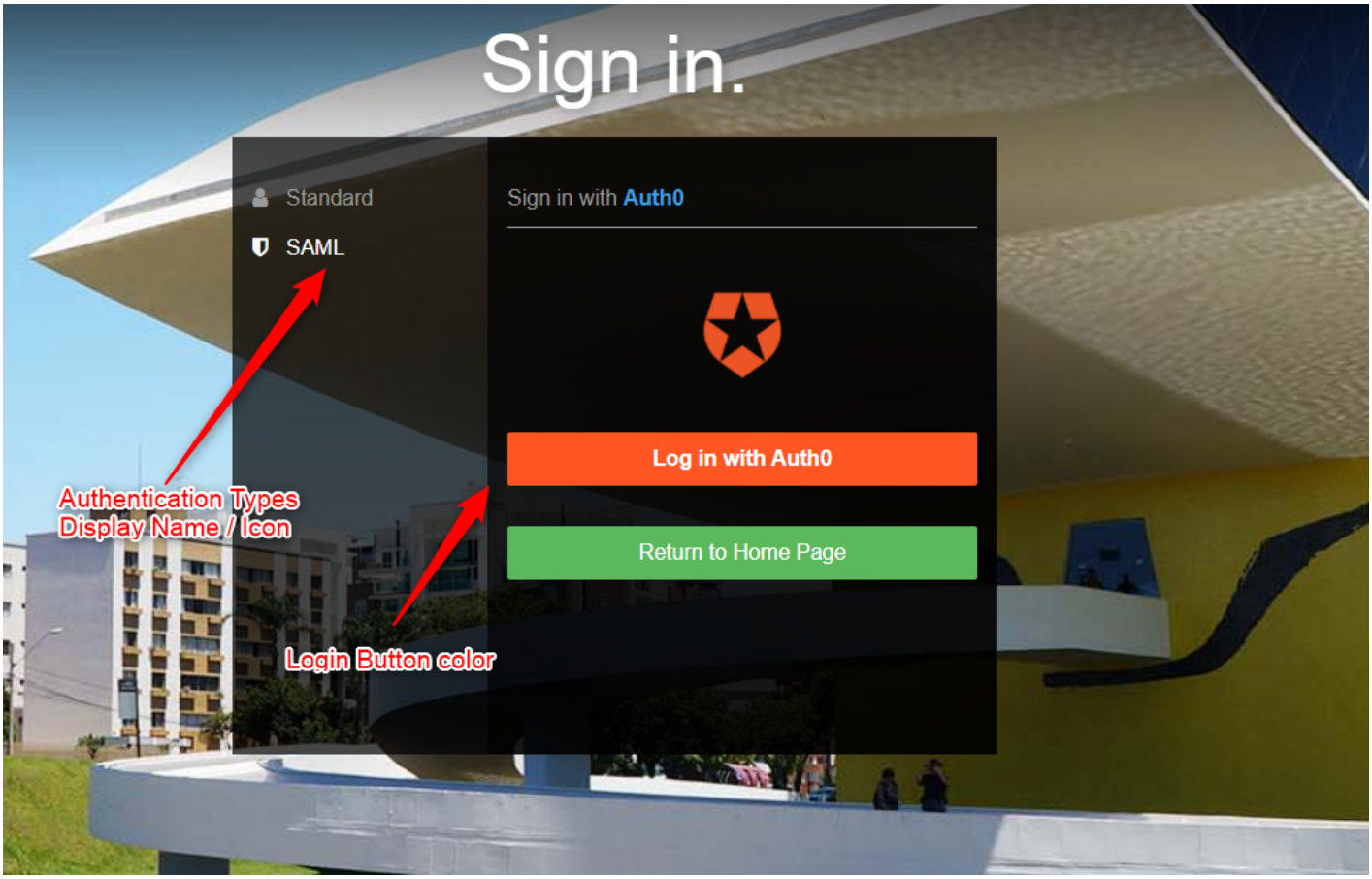
Default After Login Redirect Url

When an unsolicited SSO response arrives at the Service Provider , the user (if authenticated) is redirected to this default URL.

> The URL to which SP redirects successfully authenticated end users.



You can customized login page



Advanced

Authentication Types

	ID	Handle	Display Name	
	1	concrete	Standard	
	2	community	concrete5.org	
	3	facebook	Facebook	
	4	twitter	Twitter	
	5	google	Google	
	6	external_concrete5	External concrete5	
	30	Saml	SAML	

Use Custom Certificate

An X.509 certificate and associated private key are required if SAML messages sent by our service provider (SP) are to be signed (When the **Sign Request** option is enabled) .

Certificate is published with your SAML metadata and is freely distributed to your relying parties. Private key, just as it’s name says, should remain private and for your eyes only. Due to security issues, certificates expire after some time, and you have to renew them in order to keep SAML signing working.

By default, Our SP uses the tenant private key to sign SAML requests (When the **Sign Request** option is enabled).We recommend to provide your own credential key pair to ensure secure data transfers with identity providers.

The steps below are an example of the process for generating a public/private key pair for key exchange, using OpenSSL. To execute the following commands, you will need an OpenSSL runtime installed (which you can download and install from the OpenSSL website, or install one from your operating system’s package management system) .

1) You can generate your own certificate and certificate using this command :

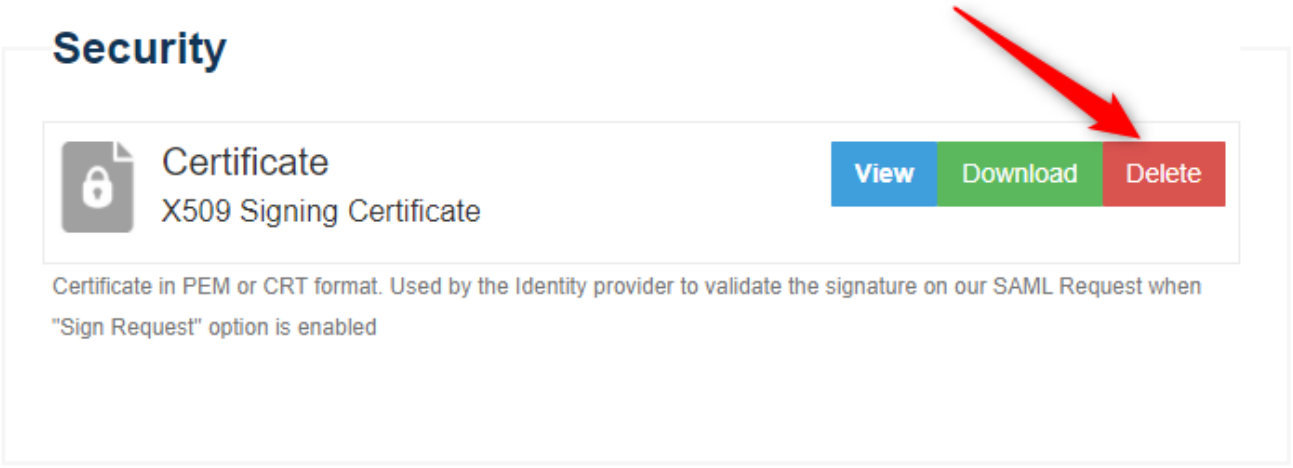
```
openssl req -new -x509 -days 365 -nodes -sha256 -out sp_certificate.crt -keyout sp_private.key
```

2) Provide information at each prompt .

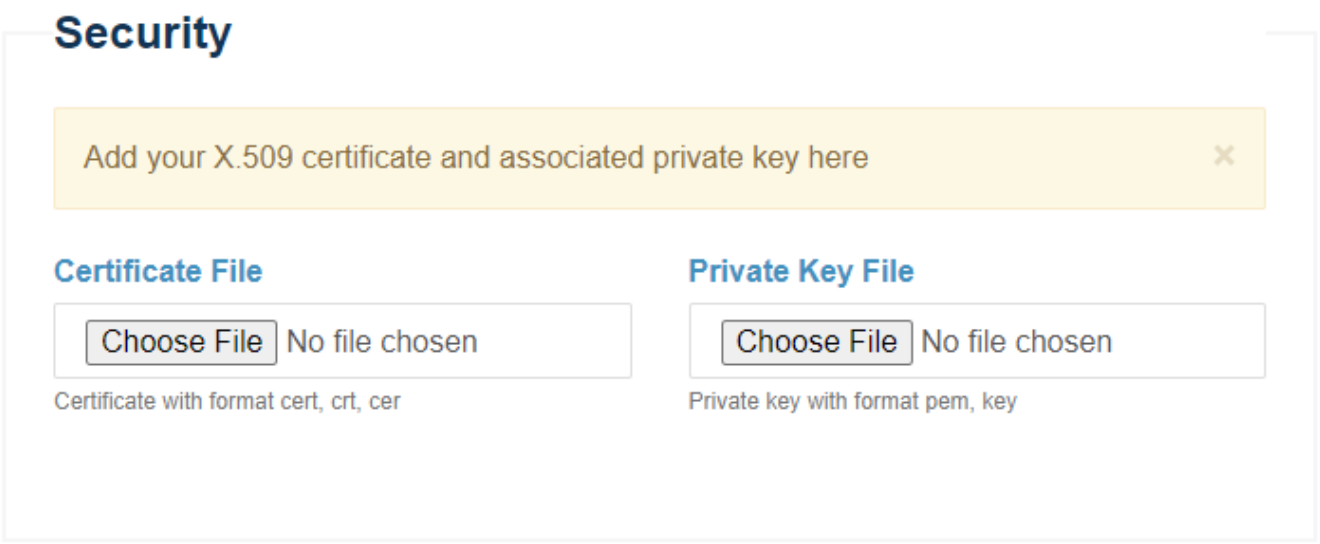
> Two files, **sp_certificate.crt** and **sp_private.key**, are created in the directory where you ran the command .

3) Go to " *Dashboard > SAML Service Provider > Configuration and Settings* " page .

4) In **Security** section, click **Delete** on the certificate you want to delete. The **Delete Certificate** window displays. Click **Yes** to confirm. Otherwise, click **No**.



5) Add your last generated files here



6) Click on **Save**

> Use certificates with strong cryptographic keys for digitally signing or encrypting SAML messages, and renew or replace the certificates every three to five years.

> Neither the private key file nor its password should be shared with third parties .

Make sure that all of your certificates are valid, and have not expired or been revoked.

Enabling signed requests requires that the IDP be updated whenever the signing certificate used by the SP is renewed or replaced.