Gigya as SAML IdP

Note: Gigya as SAML IdP is a premium Gigya platform that requires separate activation and utilizes Gigya's Registration-as-a-Service (RaaS). If these are not yet a part of your existing site package, please contact Gigya Support via the Support link in the top menu of your Console Dashboard or email support@gigya.com.

Introduction

Gigya's SAML IdP service enables identity federation, on-top of Gigya's Customer Identity Management, via the SAML2.0 protocol. The SAML IdP makes it possible for any service providers that support SAML 2.0 to authenticate with Gigya's Customer Identity management.

This functionality is supported for customers using Gigya's Registration-as-a-Service.

In this guide Gigya functions as an IdP, which means that Gigya provides an SSO login (or logout) to a third-party service provider. The SAML protocol supports different profiles and binding options, and Gigya supports the web browser SSO profile, with HTTP POST and HTTP Redirect binding.

This guide walks through Gigya's SAML Identity Provider (IdP) setup and serves as a reference document for the configuration options.

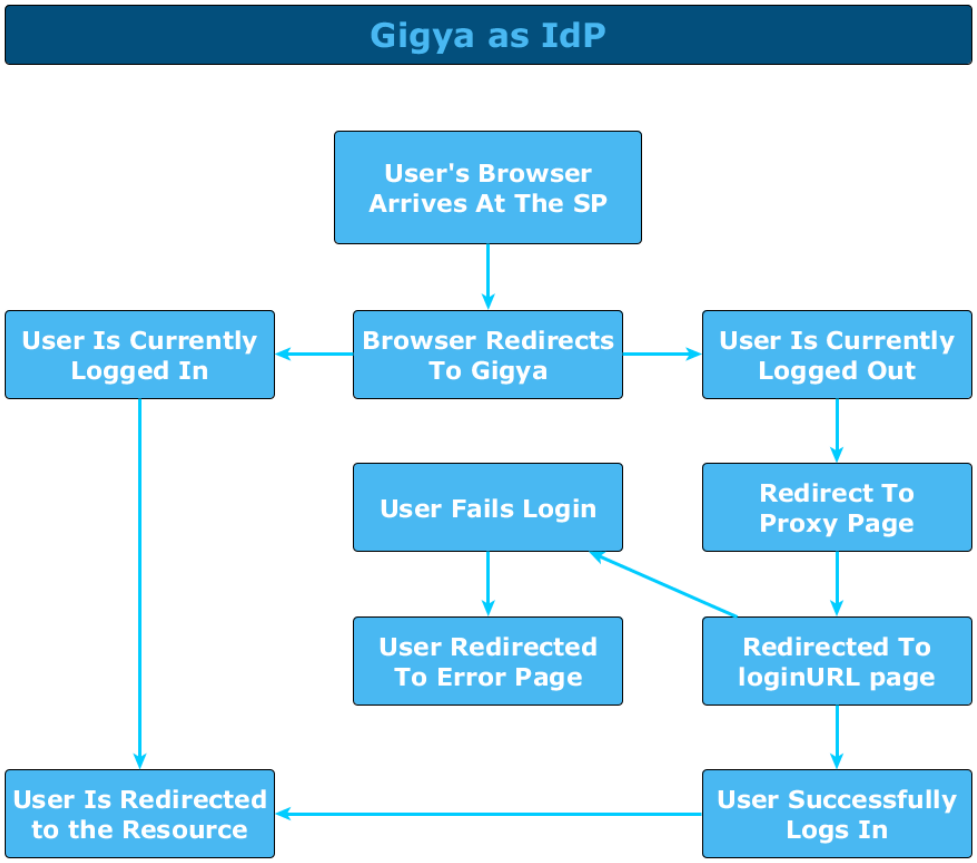
SAML is not supported natively in iOS but does work in WebView.

Unable to render {include} The included page could not be found.

Watch an Instructional Video

To watch a video about this subject, you can visit our Enablement portal with your approved SAP customer or partner ID (S user). Please visit the About page to find out how to get an S user.

Basic Login Flow



Specific SSO and SLO Flows

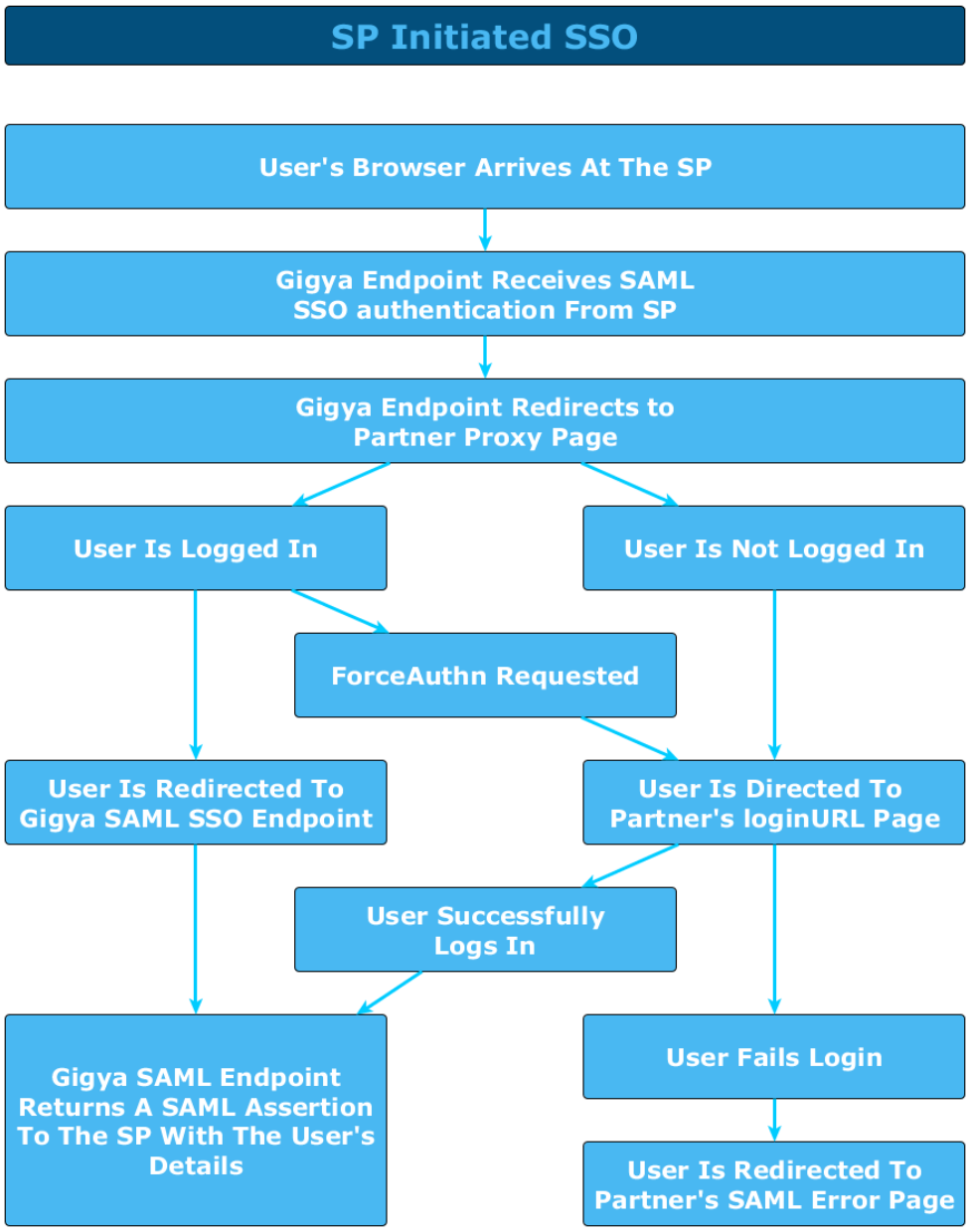
There are two Single Sign-On (SSO) and two Single Log Out (SLO) flows that Gigya supports, in all four Gigya is the IdP.

SSO

1. SP initiated SSO:

1. The Gigya SAML endpoint receives a SAML SSO authentication request from an SP to the SAML SSO endpoint.
2. The Gigya SAML endpoint redirects to a configured Gigya proxy page on the partner site.
3. If the user is not logged in, or if **ForceAuthn** is requested, the user is redirected to the configured login page on the partner site (the proxy page holds the login page URL).
4. Once the user is logged in, he should be redirected back to the Gigya SAML SSO endpoint via the proxy page.
5. The Gigya SAML endpoint returns a SAML assertion to the SP with the user details.

* isPassive() is A SAML property indicating that no login UI should be presented to the user and may be used within the above flow if you simply want to send the completed SAML Assertion to the SP. Note that this will return an empty Assertion if the user is not logged in.



2. IdP initiated SSO:

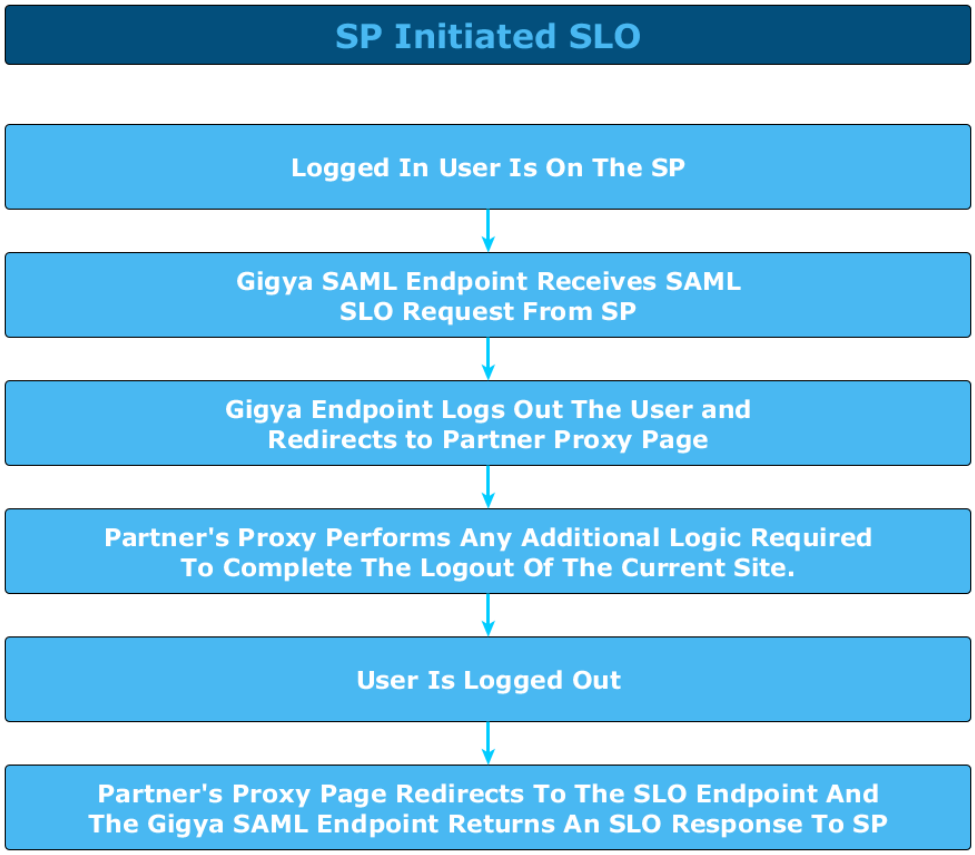
The user needs to be already logged in for IdP initiated SSO.

1. The SSO can be initiated from the JavaScript SDK or by redirecting to the SSO endpoint with the IdP initiated SSO parameters.
2. The Gigya SAML endpoint generates an authentication response and redirects to the SP ACS endpoint.

SLO

3. SP initiated SLO:

1. The Gigya SAML endpoint receives an SLO request to the SAML SLO endpoint.
2. The Gigya SAML endpoint logs out the user and then redirects to the proxy page on the partner site (where any additional logic should exist to complete the user's logout from the site, i.e., delete any session cookies).
3. The proxy page redirects back to the SLO endpoint and the Gigya SAML endpoint returns an SLO response to the SP.



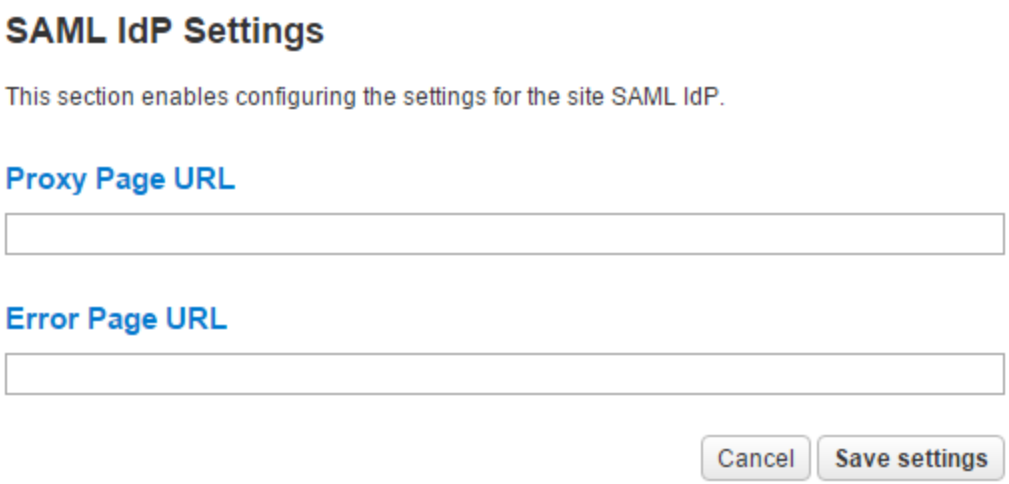
4. IdP initiated SLO:

1. When calling accounts.logout using Gigya's Web SDK, Gigya will initiate SAML SLO to all connected SPs.

Partner Integration for IdP

The following are pages required for the SAML protocol, which need to be configured by the partner and specified in the SAML IdP Settings page of the Gigya Console.

- SAML Proxy Page
- SAML Error Page



Partner Steps for Integrating SAML IdP

1. The partner creates the Gigya HTML Proxy page:
    - Set the login and logout page URL parameters in the proxy page.
    - Place the proxy page in an accessible URL in the partner site.
    - Include the gigya.fidm.saml.continueSSO() method on the login page to redirect after a successful login.
2. Configure the Gigya IdP parameters:
    - Proxy page URL
    - SAML error page URL (optional)
3. Configure the SPs that are allowed to SSO with the site manually or, when possible, by using fidm.saml.importIdPMetadata API.

SAML Objects

Objects:

- ExternalSP - External SP configuration parameters.
- GigyaConfig - Gigya SAML configuration parameters.

IdP Setup

The following guide walks through Gigya's SAML Identity Provider (IdP) setup and serves as a reference document for the configuration options.

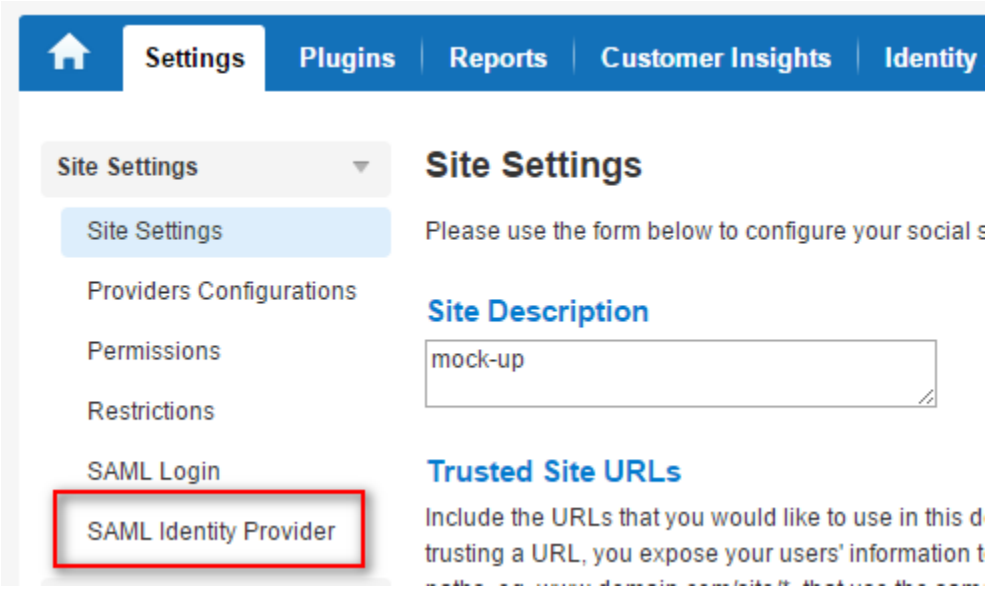
Introduction

The SAML IdP section of Gigya's site includes:

- The main SAML IdP page, which contains links to the configuration of the SAML IdP settings and to the SAML IdP Metadata, and also displays a SAML SPs table.
- The Configure SAML IdP Settings form, where you define the Proxy and Error Page URLs.
- The SAML IdP Metadata page, which contains the configuration data for the external SP.
- SAML SP Settings, available via clicking on any existing SAML SP, or the Edit icon.
    - The connected page includes all the settings for the SP as well as the Add Attribute and Edit Attribute windows.
- An Import SP window, for importing an SP's settings from a Metadata URL.

Go to the SAML IdP section of Gigya's website. Please make sure you are signed in.

The SAML IdP page may also be accessed by clicking **Settings** in the upper menu and then **SAML Identity Provider** in the left menu:



SAML Identity Provider

This page links to the configuration of the SAML IdP settings, and also to the SAML IdP Metadata page. The metadata includes information about the IdP, which is Gigya in this case.

You may add or import an unlimited number of SPs. Once you configure an SP you can edit and/or delete it.

This page displays the SAML SPs table.

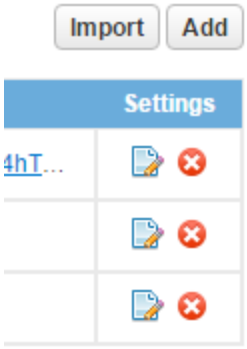
Configure SAML IdP Settings

SAML IdP Metadata

SAML SPs Import Add

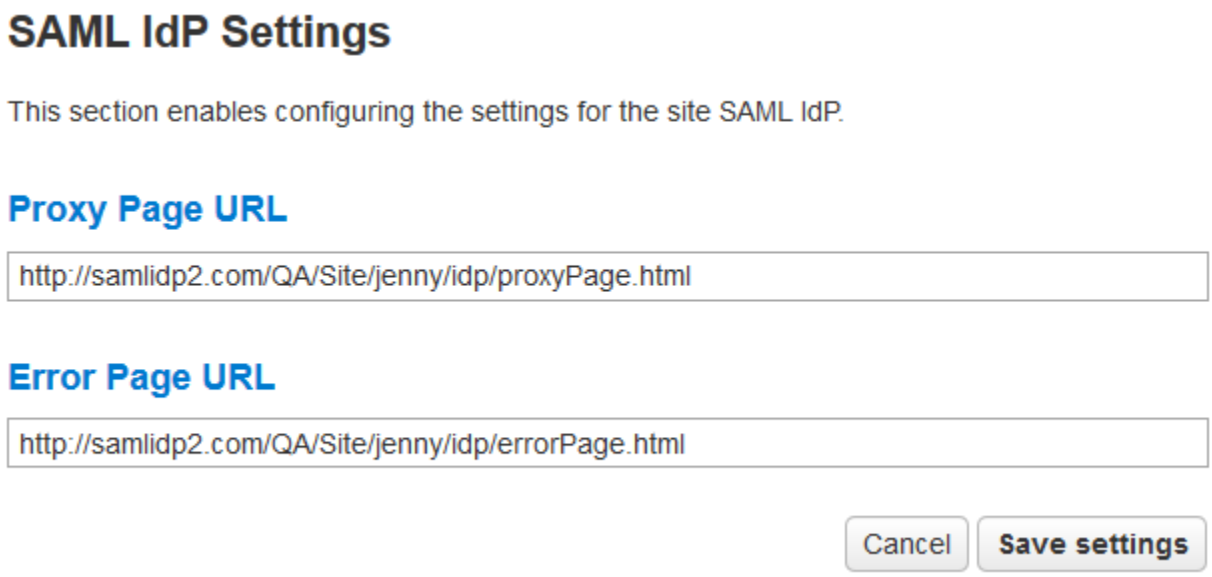
Name	Issuer	Settings
gigyaInHouse	https://fidm.gigya.com/saml/v2.0/3_6QwRZiDAP8OM9IbdNqLiOfkd15BtfVaMjfBTalghLYGpjphw9WA54hT...	
salesForce	https://[redacted]my.salesforce.com	
salesForce2	https://[redacted]my.salesforce.com/	

When you click the Add button or Edit icon, the SAML SP Settings form is displayed.



Configure SAML IdP Settings

This section is used to configure the settings of the site SAML IdP.

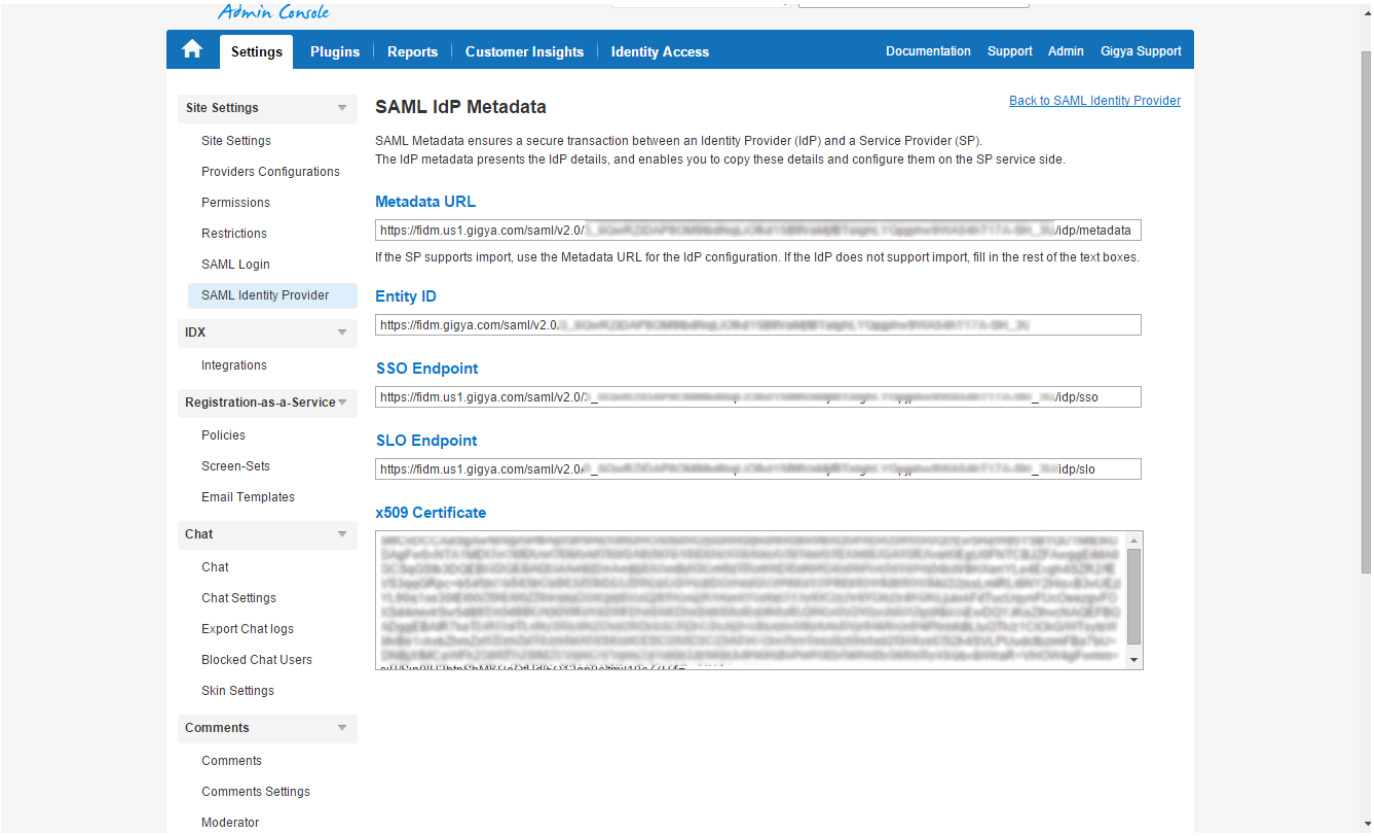


The input fields are as follows:

- Proxy Page URL - an HTML page that the partner places in his site in order to use the Gigya SAML IdP.
- Error Page URL - a URL for an error page to display errors.

SAML IdP Metadata

The main SAML Identity Provider page has a link to SAML IdP Metadata:



This metadata ensures a secure transaction between the Identity Provider (IdP) and the Service Provider (SP).
The IdP metadata presents the IdP details, and enables you to copy these details and configure them on the SP service side.

SAML SP Settings With Gigya As IdP

In order to act as a SAML IdP, you must configure approved Service Providers (SPs), enabling customers to provide SAML IdP services.
In the main SAML Identity Provider page, the SAML SPs table is displayed. You can click "Add" to add a new SAML SP, or "edit" to edit one of the existing SAML SPs.

Adding or editing a SAML SP opens the following SP settings form:

Name ?

[redacted] *

Issuer ?

[redacted] *

Session Lifetime (Minutes) ?

1440

Assertion Consumer Service URL ?

[redacted] *

Single Logout Service URL ?

[redacted]

Single Logout Service Binding ?

urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST ▼

Name ID ?

- (•) Auto-generated Unique ID per SP (Persistent Pseudonym) ?
- ( ) Map profile field ?

Profile field type ?

Unspecified ▼ | urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified

- [x] Sign Authentication Request ?
- [ ] Encrypted Assertion ?
- [x] Signed Assertions ?

Attribute Map ? Add attribute

Profile Field	Attribute	Settings

x509 Certificate ?

[redacted]

x509 Encryption Certificate ?

[redacted]

The input fields are as follows:

- **Name** - A custom name for this External Service Provider. This field must be unique within the site (Gigya Console).
- **Issuer** - The Service Provider's entity ID.
- **Session Lifetime (minutes)** - The time span for which SAML sessions are valid.
- **Assertion Consumer Service URL** - The SP's ACS URL.
- **Single Logout Service URL** - The SP's SLO URL.
- **Single Logout Service Binding** - The type of SLO binding.
- **Name ID** - Define whether to auto-generate a unique ID for the SP, or define it manually by mapping to it*.
- **Sign Authentication Request** - Specifies whether SAML requests should be signed.
- **Encrypt Assertion** - Specifies whether SAML assertions should be encrypted.
- **Signed Assertion** - Specifies whether SAML assertions should be signed.
- **Attribute Map** - A table that maps attribute names between the SP and the IdP.
- **x509 Certificate** - The IdP x509 certificate.
- **x509 Encryption Certificate** - The x509 certificate that is used to encrypt the assertions.

* **Name ID** - Once **Name ID** is set, all users that log into the SP will be associated with a variation of this unique ID. Replacing or changing this mapping at any point in the future will prevent the SP from associating users with their previously existing data *and may lead to unauthorized access of a given user's personal data* if their previous nameID is ever reassigned to a different user.

Recommended Best Practices:

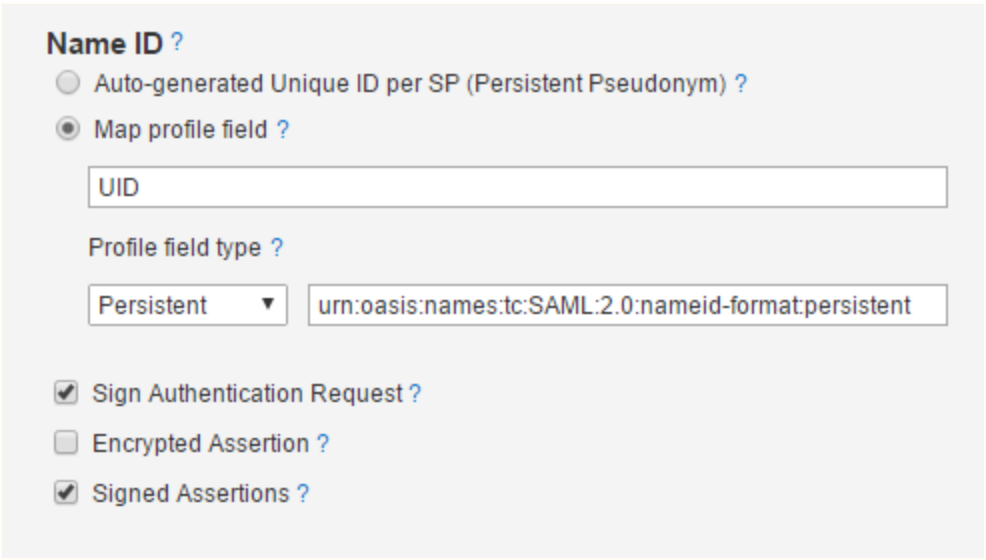
- **Auto-generated unique ID per SP (Persistent Pseudonym)**
  - This option should be used by an IdP that wants to prevent SPs from sharing data across accounts.
  - However, since the ID is derived from the site ID and IDs are not portable between sites, accounts exported from one IdP to another will be treated as a new account on the user's next login to the SP.
- **Map profile field**
  - This option should be used in all other cases.

SSO Group Considerations

When using Multiple SPs within a Gigya SSO Group with a Gigya IdP:

- All SPs within the SSO Group must use the exact same IdP configuration, assuring they all refer to the IdP by the same name.

- In the **Name ID** section you must:
  - Select the **Map profile field** option and set the field to **UID**.
  - Select **Persistent** for the Profile field type.



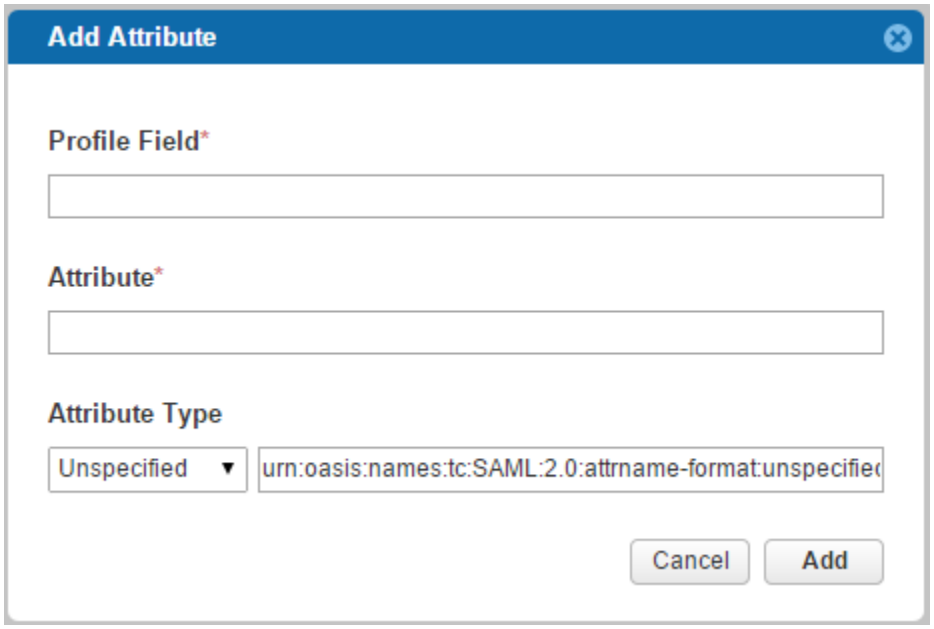
This ensures that your users are always identified as the same user between the multiple SPs.

Adding SAML Attributes

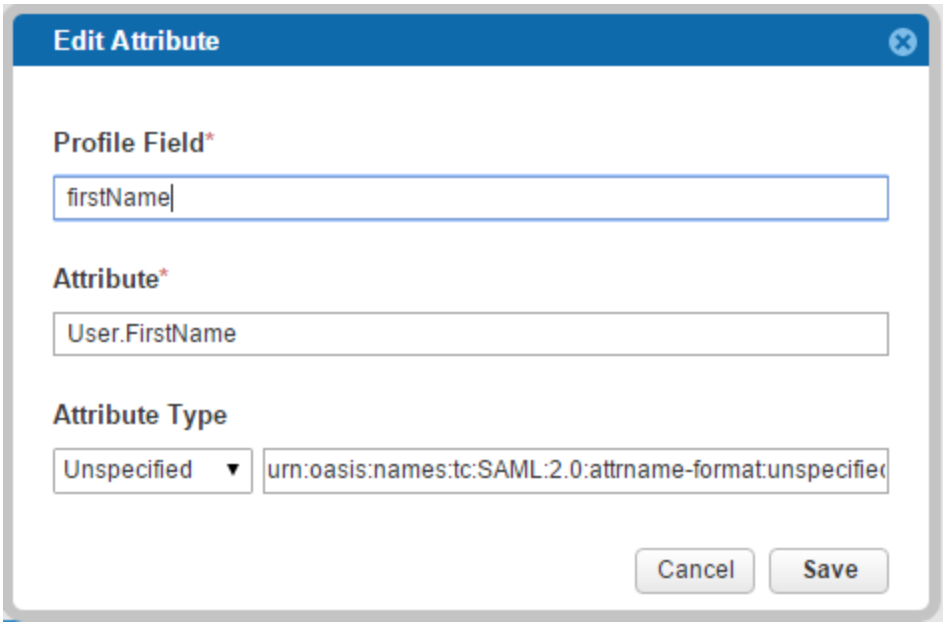
You can add attributes to the attribute map by clicking the **Add attribute** button on top of the table.

> Changes will be saved only if you click the **Save Settings** button on the parent page.

When you click **Add**, the **Add Attribute** window will open:



When you click **Edit**, the **Edit Attribute** window will open:



In the above example the Gigya **profile.firstName** field is mapped to the **User.FirstName** attribute name of the external SP.

Supported Attributes

Profile Field can be any of the following Gigya supported attributes with or without the **profile.** prefix:

> These properties are case-sensitive and must be used exactly as described below.

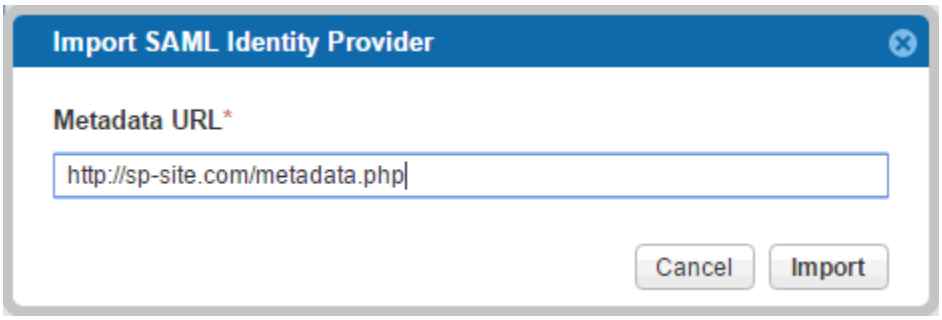
- firstName
- lastName
- nickname
- photoURL
- profileURL
- address
- thumbnailURL
- email
- country
- state
- city
- zip
- gender
- age
- birthDay
- birthMonth
- birthYear
- loginProvider
- UID
- uid
- account.isVerified
- data.*customField*
  - This attribute allows you to define custom attributes i.e., data.myCustomField.

For information on determining **Attribute** names to map supported ID's to, we suggest reviewing the Shibboleth reference maintained by The Wharton School at the University of Pennsylvania.

Import SAML SP

Unable to render {include} The included page could not be found.

In the main SAML IdP page, above the SAML SPs table, there is an "Import" button, which opens the "Import SAML SP" window:



In order to import a SAML SP configuration, you must provide its metadata URL.

Passing Data in Custom Parameters

It is possible to pass data from the SP to the IdP Proxy Page using custom parameter names. To achieve this, simply add the additional parameters to the request URI, e.g.,

C-Sharp Example

```
using System;
using System.Collections.Generic;
using System.IO;
using System.IO.Compression;
using System.Linq;
using System.Net.Http;
using System.Text;
using System.Threading.Tasks;
using System.Web;

namespace SamlDemo
{
    class Program
    {
        static void Main(string[] args)
        {
            const string DATA_CENTER = "us1.gigya.com"; //
https://developers.gigya.com/display/GD/Finding+Your+Data+Center
            const string SP_API_KEY = "The SP API Key";
            const string IDP_API_KEY = "The IdP API Key";

            var samlRequestString = $@"<samlp:AuthnRequest
                                ID='{Guid.NewGuid()}'
                                Version='2.0'

IssueInstant='2019-01-31T12:22:15.636Z'

Destination='https://fidm.{DATA_CENTER}/saml/v2.0/{IDP_API_KEY}/idp/sso'
```

```
                                    ForceAuthn='false'
                                    IsPassive='false'

AssertionConsumerServiceURL='https://fidm.{DATA_CENTER}/saml/v2.0/{SP_API_
KEY}/sp/acs'

xmlns:samlp='urn:oasis:names:tc:SAML:2.0:protocol'>
                                        <saml:Issuer
xmlns:saml='urn:oasis:names:tc:SAML:2.0:assertion'>https://fidm.gigya.com/
saml/v2.0/{SP_API_KEY}</saml:Issuer>
            <samlp:NameIDPolicy
Format='urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified'
AllowCreate='true' />
                                </samlp:AuthnRequest>";

            var builder = new UriBuilder();
            builder.Scheme = "https";
            builder.Host = $"fidm.{DATA_CENTER}";
            builder.Path = $"saml/v2.0/{IDP_API_KEY}/idp/sso";
/*HERE ->*/ builder.Query =
$"SAMLRequest={HttpUtility.UrlEncode(Deflate(samlRequestString))}&customPa
rameter01=some_data"; // your custom params

            var client = new HttpClient();
            var response = client.GetAsync(builder.Uri.ToString()).Result;
        }

        static string Deflate(string input)
        {
            byte[] bytes = Encoding.UTF8.GetBytes(input);

            using (MemoryStream output = new MemoryStream())
            {
                using (DeflateStream gzip = new DeflateStream(output,
CompressionMode.Compress))
                    gzip.Write(bytes, 0, bytes.Length);
                var base64String =
Convert.ToBase64String(output.ToArray());
                return base64String;
            }
```

```
            }
        }
    }
```

Additional Information

> Any unique user account can only be connected to a single custom SAML IdP or OIDC OP. Once connected, the user will not be able to use a different SAML/OIDC provider if they exist. However, a user can be associated with both a SAML IdP and an OIDC OP concurrently.

For instructions on converting the Gigya x509 certificate to a text file, please see here.

When calling accounts.logout in an SSO group, it is important to note that the callback initiates when the response is received to the current site. This means that it is possible that the logout event has not completed on all other child sites. If you require that the user is logged out from all child sites prior to initiating the callback (i.e., when redirecting the user), you must set a timeout inside the callback of accounts.logout.

> If using SAML and you want to enable the ability to **Link Accounts**, or **socialize.addConnections** you must set ***federation.allowMultipleIdentities*** to **true** using accounts.setPolicies. Read these **important notes** before enabling the **allowMultipleIdentities** parameter.