Using SAML for Single Sign-On in the SOA Software Platform

SOA | software™

Policy Manager / Community Manager

Using SAML for Single Sign-On in the SOA Software Platform
Version 7.2
December, 2014

Copyright

Trademarks

SOA Software, Inc.

SOA Software, Inc.
12100 Wilshire Blvd, Suite 1800
Los Angeles, CA 90025
(866) SOA-9876
www.soa.com
info@soa.com

Disclaimer

Contents

Using SAML for Single Sign-On in the SOA Software Platform

This document provides information on the SOA Software platform's support of the SAML Web Browser SSO Profile, Service Provider role, used to provide single sign-on services for user login in various applications in the platform. It includes:

- A brief overview of SAML as it relates to the SOA Software Platform implementation
- Information about how the platform offers support of the SAML Web Browser SSO profile
- Instructions for setting up SAML Web Browser SSO profile support
- Sample requests, responses, and metadata
- Identity Provider configuration examples for SSO Circle and PingFederate
- Troubleshooting
- Glossary of terms

Chapter 1 | Overview of SAML

SAML, the Security Assertion Markup Language, is an XML-based identity federation standard. Among other capabilities, SAML can be used for single sign-on.

SAML is used for exchanging authentication and authorization data between a Service Provider (providing a service to the user) and an Identity Provider (providing identity verification of the user to the Service Provider).

The SOA Software platform supports SAML Web Browser SSO for two key purposes:

- **Single sign-on**: As the single sign-on token (along with other tokens such as OpenID Connect's id_token token) as part of the Community Manager Developer Portal login.
- **OAuth Provider domain**: For resource owner login when using OAuth Authorization Server.

Why Choose SAML Web Browser SSO?

SAML and OpenID Connect are both very popular and mainstream standards that support single sign-on. OpenID Connect is essentially JSON-based, whereas SAML is an XML implementation.

There are two main reasons why you might choose SAML over OpenID Connect for your Policy Manager/Community Manager implementation:

- An existing SAML implementation
- The need to support webservices, REST APIs, and user login with a common infrastructure

OpenID is another single sign-on solution still in use. However, OpenID has been deprecated by Google in favor of OpenID Connect; in addition, OpenID Connect is more flexible and more REST API-friendly. The platform supports OpenID for backwards compatibility, but we do not recommend adopting this standard.

How SAML Web Browser SSO Profile Works: High-Level View

As with other SSO solutions, SAML provides authentication and verification of end-users via the SAML Web Browser SSO Profile, so that apps can easily outsource this critical, sensitive, and complex task. At a high level, the exchange of information is as follows:

1 The end-user requests a service from an app.

   App = Service Provider; provides service to the end-user. Corresponds to Relying Party role in OpenID Connect; relies on the Identity Provider for verification of the user's identity.

2 Before providing the service to the user, the app must authenticate the user. To do this, the app redirects the user to a supported SAML Identity Provider (IdP) of the end-user's choosing. The IdP:

   a) Requests authentication information from the user.
   b) Verifies the information.
   c) Logs the user in.
   d) Redirects the user back to the app.

   The Identity Provider provides user authentication services to the Service Provider. This role corresponds to the Asserting Party role in OpenID Connect. The Identity Provider sends the authentication information in the form of an XML-based SAML Assertion.

3 The app delivers the service to the end-user.

Of course, before being able to authenticate their users via a SAML Identity Provider, the app must first set up an account with the SAML Identity Provider. In addition, for authentication to be successful, the end-user must have an account with the SAML Identity Provider.

In the SOA Software Platform, Policy Manager/Community Manager acts as a SAML Service Provider. Configuring this solution requires coordination between values set up in your account with the SAML Identity Provider and in the domain setup in Policy Manager so that messages can be sent and received between Policy Manager and the applicable Identity Provider.

How SAML Web Browser SSO Works: Behind the Scenes

Part of the SAML Web Browser SSO Profile standard includes a metadata file which includes information and settings that allow the Service Provider and the Identity Provider to validate each other's messages.

Once you've chosen your SAML Identity Provider, you create a Service Provider account. Since the platform is acting as your Service Provider, some of the values you specify are determined by your own choices and some values are determined by the platform and what it supports. Once you provide values and make choices, the Identity Provider generates a metadata XML file that includes values relevant to messages from the Identity Provider to the platform (as your Service Provider).

In the same way, you create a domain in Policy Manager for the Identity Provider. Here, you provide values relevant to the Identity Provider and values you specify with the Identity Provider, such as the attributes you will use to identify your users with the Identity Provider. For example, you might use firstname, lastname, and email address, or you might use username and password. Once you've set up this information, the platform generates a metadata XML file that includes values relevant to messages from the Identity Provider to the Service Provider.

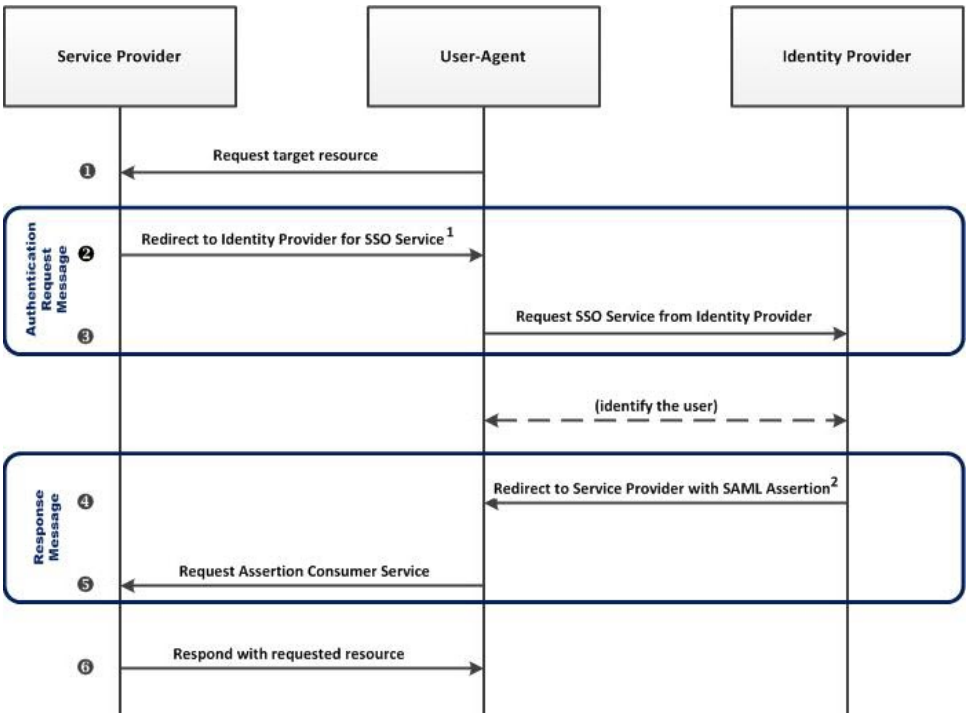
This exchange of information via metadata sets the groundwork for the establishment of mutual trust and secure exchange of information between the two parties. Validation, encryption, and decryption go on in the background and are transparent to the user, as shown in the diagram below.

In some cases, such as with SSO Circle, the IdP publishes a generic metadata file. In other cases, such as PingFederate, the IdP metadata file is customized for each account. The Service Provider metadata file is always custom; when configuring your account with the IdP you must provide the metadata or configure the values manually.

SAML Web Browser SSO: Process Flow Diagram

The sequence diagram below shows the basic exchange of information between the consumer (via the User-Agent), the Service Provider, and the Identity Provider when the SAML Web Browser SSO profile is used for single sign-on.

Note: The diagram below is general to SAML. For specific options supported, refer to Supported SAML Bindings for Single Sign-On on page 10.



In the above:

1 **Redirect to Identity Provider for SSO Service**: Service Provider sends <**AuthnRequest**> request (authentication request) to Identity Provider. Three bindings offered by the SAML specification:
   - HTTP POST. Sends the message content as a POST parameter. For more information, see HTTP POST on page 57.
   - HTTP Redirect. Redirects the user to the Identity Provider for login. Sends the message content in the URL. For more information, see HTTP Redirect on page 57.
   - HTTP Artifact (not currently supported by the SOA Software solution). Instead of sending the message content, sends a SAML Artifact to the content so the Identity Provider can retrieve it from an Artifact Resolution Service. For more information, see HTTP Artifact on page 57.

2 **Redirect to Service Provider with SAML Assertion**: Identity Provider sends <**Response**> message to Service Provider. Two bindings offered by the SAML specification:
   - HTTP POST
   - HTTP Artifact

SAML Specifications

Below are links to information relating to the SAML 2.0 specifications:

- SAML specification: http://saml.xml.org/saml-specifications
- SAML assertions: http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf
- SAML bindings, including HTTP Redirect, HTTP Artifact, and HTTP POST: http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf
- SAML Glossary: http://docs.oasis-open.org/security/saml/v2.0/saml-glossary-2.0-os.pdf

Chapter 2 | SAML Web Browser SSO Support in the SOA Software Platform

This section provides information specific to the SAML implementation in the SOA Software platform. It includes:

Supported Features

The SOA Software Platform version 7.2 supports single sign-on with SAML as the token for authentication (Web Browser SSO Profile) in the following scenarios:

1. To authenticate developers in the developer portal.
2. To authenticate end-users when issuing OAuth grants.

Supported SAML Bindings for Single Sign-On

Policy Manager/Community Manager supports the following SAML bindings for Service Provider-initiated single sign-on in Community Manager version 7.2:

For authentication request:

- HTTP Redirect
- HTTP POST

Note: this solution currently does not support HTTP Artifact for authentication request messages.

HTTP Redirect sends the full authentication request as a query parameter, whereas HTTP POST sends the information as a POST parameter, in the payload.

The Service Provider cannot do a redirect with POST, so with HTTP POST the Service Provider returns an HTML form, and when the form is loaded into the browser it submits the form information to the Identity Provider.

Tip: One reason you might choose to go with HTTP POST rather than HTTP redirect is because of limitations in the length of the redirect URL. Signing of the SAML authentication request adds to the length of the message, and if the URL is too long it can cause problems.

For response (issuing the SAML assertion):

- HTTP POST
- HTTP Artifact

HTTP POST sends the full response as a POST parameter, in the payload. HTTP Artifact sends the artifact as a query parameter; the artifact is a handle for the full response.

In order to use HTTP Artifact binding in response messages, you must set up an artifact resolution service (ARS) with the IdP.

Both bindings are secure. However, HTTP Artifact is more secure, because in order to get the entire message (SAML Assertion) the service provider must validate again with the sender in a synchronous exchange after receiving the artifact reference, to access the full artifact via the artifact resolution service.

Currently Not Supported

The SOA Software Platform currently does not support the following:

- IdP-initiated SSO
- SP-initiated SLO (single logout)
- IdP-initiated SLO
- HTTP Artifact binding for authentication of the request from the Service Provider to the Identity Provider.

SAML Version

The SOA Software solution for using SAML for single sign-on in the SOA Software Platform supports SAML Version 2.0.

Supported Identity Providers

The SOA Software Platform SAML single sign-on feature should work with any SAML Identity Provider that supports SAML Web Browser SSO Profile for Service Provider-initiated SSO. It has been tested with the following:

- SSOCircle: see http://www.ssocircle.com
- PingFederate: see https://www.pingidentity.com
- OpenSSO (now less popular) and a more recent product that builds on OpenSSO, OpenAM by Forgerock

This document gives general instructions applicable to any Identity Provider, and provides a couple of examples of setup for supported Identity Providers.

Chapter 3 | Setting Up the SAML Web Browser SSO Feature

This section provides information about setting up the SAML Web Browser SSO feature, including planning, installation, setup steps in Policy Manager and Community Manager, and testing.

Note: Once you have everything set up, any change made in the SOA Platform that affects the URLs used for sending and receiving messages related to SAML will require update in the Service Provider account configuration with the Identity Provider. For information on updating, see Modifying an Existing SAML Installation on page 50.

Requirements for the SAML Domain to Work in Community Manager

At a high level, to set up a domain that uses SAML Web Browser SSO to provide single sign-on services in Community Manager, you must complete the steps below.

Note: This document assumes that you've already created an account with your selected Identity Provider.

To set up the SAML Web Browser SSO feature: high-level procedure

1. Install the plug-ins in relating to the SAML Web Browser SSO feature in the correct containers. See Step 1: Install the Plug-Ins to Support SAML Web Browser SSO on page 13.
2. Gather the information you'll need to provide in the configuration steps. See Step 2: Gather Information on page 13.
3. Determine setup sequence. Whether it's best to do the Policy Manager setup first, or the Identity Provider setup first, depends on which Identity Provider you are using. See Step 3: Determine Setup Sequence on page 14.
4. In Policy Manager, set up the SAML domain, with the values relating to your SAML installation (gathered in Step 2). See Step 4: Configure the Domain in Policy Manager on page 14.
5. In your Identity Provider account, set up the Service Provider connection, using the values you gathered in Step 2. The specific steps will vary according to the Identity Provider you're using. Examples for specific IdPs are given later in this book:
   - Identity Provider Configuration Example: SSO Circle on page 35
   - Identity Provider Configuration Example: PingFederate on page 42

   Note: Depending on your specific setup and the Identity Provider you are using, it might be more efficient to do the setup in the Identity Provider (Step 4) first. In either case, the important thing is to make sure that the applicable values are set up correctly in both places.

6 In Community Manager, complete the setup by following the applicable procedure, depending on how you will be using the SAML domain:
   - As a login domain: see To enable a SAML login domain in Community Manager on page 18.
   - As an OAuth provider domain: see To configure a SAML OAuth Provider domain in Community Manager on page 18.
7 In Community Manager, test to make sure your domain that uses the SAML Web Browser SSO feature works correctly:
   - As a login domain: see Testing the SAML Domain as a Login Domain on page 19.
   - As an OAuth provider domain: see Testing the SAML Domain as an OAuth Provider Domain on page 19.

Step 1: Install the Plug-Ins to Support SAML Web Browser SSO

To get support for the SAML Web Browser SSO feature, you'll need to install the following optional plug-ins in the SOA Software Admin Console to one or more containers in your implementation:

1 SOA Software SAML 2.0 Web Browser SSO Service Provider
2 SOA Software SAML 2.0 Web Browser SSO Service Provider UI

Which plug-ins you would need to install depends on the container, as follows:

- In the Policy Manager Console container: #2 above, SSO Service Provider UI.
- In every other container (Community Manager or OAuth Provider): #1 above, SSO Service Provider.

Step 2: Gather Information

When registering with the SAML Identity Provider, you might encounter different organization or terminology, but you can expect to be asked for the following pieces of information. Before setup, gather the information below and have it ready when completing the configuration steps in Policy Manager and your Identity Provider:

- The request SAML binding you will be using (HTTP POST or HTTP Redirect).
- The response SAML binding you will be using (HTTP POST or HTTP Artifact).
- Security key information for your request messages.
- If you are using HTTP Artifact binding for the response, security key information for encryption of the artifact.
- Your SAML Service Provider Entity ID (see Entity ID on page 57). This value is determined by you, but must be set up with the Identity Provider and must be unique for that Identity Provider.

  Note: The Service Provider's entity ID along with the certificate and private key are the two things that bind both sides together so that information can be exchanged securely.

- Optional attributes, such as firstname, lastname, and email address, to be sent in the SAML Assertion. For more information, see Attributes on page 56.

- The base URL for your implementation: <protocol_scheme>://<host>:<port>. For more information, see Base URL on page 56.
- Metadata information for the Service Provider (the platform). This is created as a result of Identity System setup in Policy Manager.
- Security keys and certificates.

Step 3: Determine Setup Sequence

For the SAML Browser Web SSO feature to work, you must set up certain values on the platform side, in Policy Manager, and the same values on the Identity Provider (IdP) side, in your Service Provider (SP) account setup. These values enable the exchange of information between the SP and the IdP.

Whether it's better to do Policy Manager setup first, or do the setup at the IdP first, is determined by which IdP your installation is using. Each Identity Provider has a different user interface. In addition, there are different versions. This document includes generic instructions and also setup examples for two Identity Providers:

- Identity Provider Configuration Example: SSO Circle on page 35

  With SSO Circle, it's easiest to do the Policy Manager setup first. SSO Circle publishes a generic metadata file, so you can easily provide a link to the file, or upload it to Policy Manager, and Policy Manager prefills many of the values needed for the setup wizard. Policy Manager then generates the Service Provider metadata file, and you can then paste the contents of this file into SSO Circle.

- Identity Provider Configuration Example: PingFederate on page 42

  With PingFederate, you could do it either way. In the example, the Policy Manager setup is done first here also. However, because the metadata file is not yet available, manual configuration is required. You could choose to do the PingFederate setup first.

The sequence is not as important as the fact that the values must match on both sides.

Step 4: Configure the Domain in Policy Manager

This section provides general information about the setup steps you'll need to complete in Policy Manager, including the basic procedure and an overview of the wizard.

To configure the SAML Web Browser SSO Domain in Policy Manager

1. Log in to the Policy Manager Console.
2. Click the **Configure** tab, click **Security**, and then click **Identity Systems**.
3. Click **Add Identity System** to access the Add Identity System wizard.
4. Provide the values on each page of the wizard. The wizard takes you through the following steps:
   a) Intro page: choose identity system type (choose SAML Web Browser SSO), provide name and description.
   b) Select Identity Provider configuration method.
   c) Configure Identity Provider.
   d) Configure Service Provider.

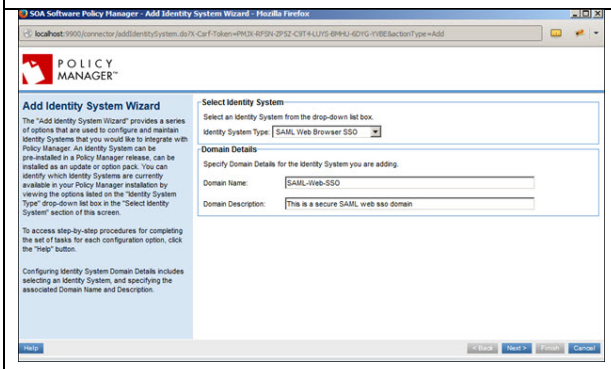
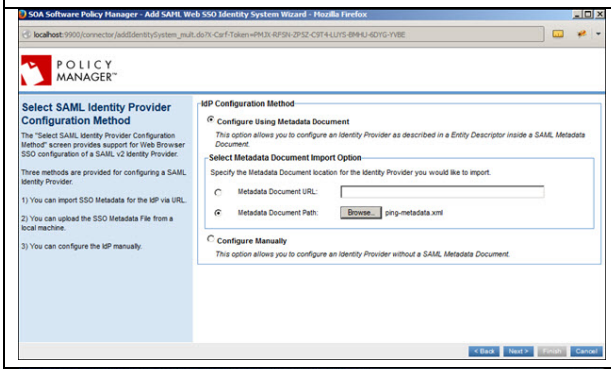
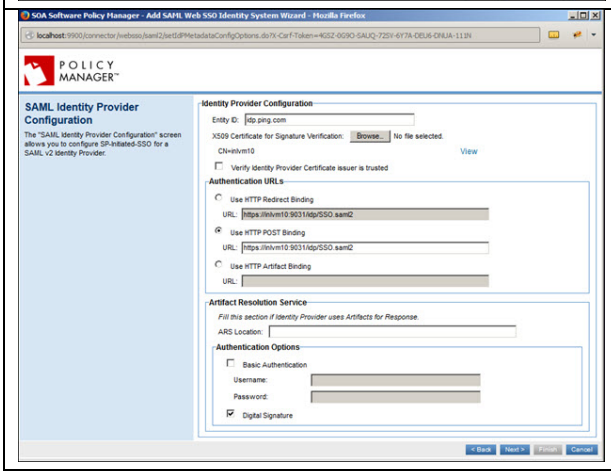
e) Define PKI keys for Service Provider.

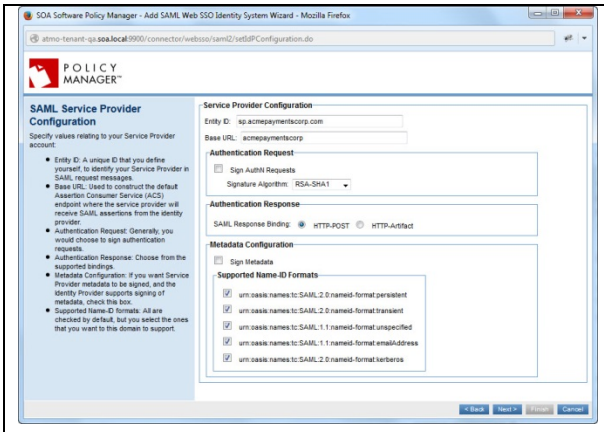
For details of the wizard and information about significant fields and values, see below.

5 Click **Finish**. The Service Provider metadata file is generated and is available at the following URL: <http(s)>://<hostname>:<port>/saml/<sp_domain_name>/metadata.

SSO Domain Configuration Wizard

The SSO Domain Configuration wizard takes you through setting up the values needed to create a SAML Web Browser SSO domain in Policy Manager. The information below is general to all Identity Providers.

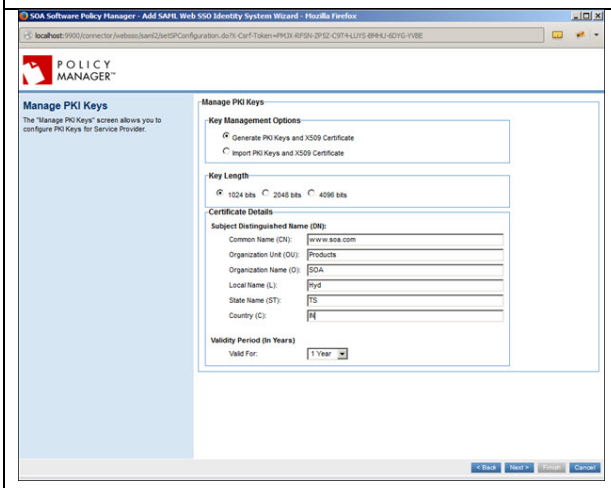
Wizard page	Details
	**Add Identity System Wizard** Provide initial values: • From the drop-down list, choose SAML Web Browser SSO • Provide a domain name and description. **Note**: the domain name is part of the path for the Service Provider metadata file Policy Manager generates when the wizard is finished. Also, the first letter of the domain name is used by Community Manager as the default for the login button. So, for example, if your IdP is PingFederate and your domain name is PingFederate domain, users will see a button labeled **P** at login.
	**Select SAML Identity Provider Configuration Method** Choose a configuration method out of the three options: • Configure by using the metadata document / URL: Provide the IdP metadata URL so the wizard can import it. • Configure by using the metadata document / Upload: Upload the IdP metadata.xml file. • Choose to configure the IdP manually.
	**SAML Identity Provider Configuration** Specify values if configuring manually, or verify if uploading metadata.xml file: • SAML Entity ID for the Identity Provider. • **Authentication URLs**: Choose the binding you will be using, either HTTP POST or HTTP Redirect, and put in the applicable URL from your Identity Provider account. • If using HTTP Artifact for the SAML response messages, details of the artifact resolution service.



Service Provider Configuration

Enter Service Provider configuration values:

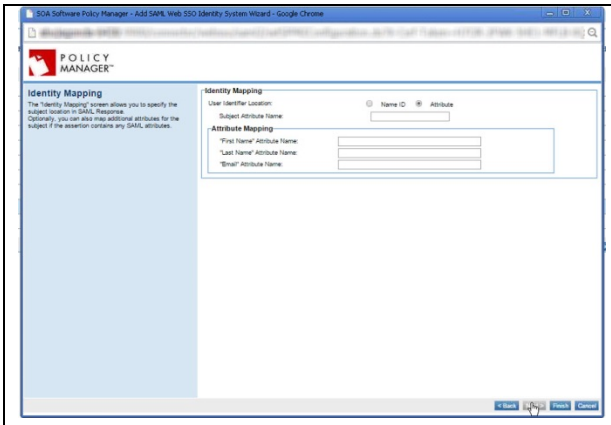
- **Entity ID**: A unique ID that you define yourself, to identify your Service Provider in the SAML authentication request messages. When setting up your account with the Identity Provider you must specify the Entity ID, which must be unique within the IdP so that the IdP can identify your Service Provider; then, you set up the same value in Policy Manager.
- **Base URL**: used to construct the default Assertion Consumer Service (ACS) endpoint, the endpoint where the Service Provider will receive SAML assertions from the Identity Provider. Must be the container address of the container where the SAML Web Browser SSO feature is initialized (<protocol_scheme>://<host>:<port>). For more information, see Base URL on page 56.
- **Authentication Request**: Generally, you would choose to sign authentication requests.
- **Authentication Response**: choose from the two supported bindings.
- **Metadata Configuration**: Choose whether or not to sign the metadata.
- **Supported Name-ID formats**: all are checked by default.



Manage PKI Keys

Set up information about the keys you will use to sign SAML authentication request messages. Outgoing messages are signed with the private key; the public key is published in the metadata file generated at the end of the wizard. The Identity Provider needs this key to verify the signature on the SAML authentication request messages.

Choose to generate or import keys. If you choose **Generate**, provide values in the **Certificate Details** section. If you choose **Import**, you'll need to choose a key management option and provide keystore details.

	**Identity Mapping** Specify attribute information: • User Identifier location: Choose whether to send the NameID as the subject of the SAML assertion, or to use an attribute: if needed, define the subject attribute name. • Subject Attribute Name: In general, the subject would be part of the NameID. However, in some cases, such as Google, the IdP does not send the subject directly. Instead, they send a unique NameID, a lengthy string. In those cases, the IdP can be configured to send the actual subject, the username, in the attribute. • Attribute Mapping: make sure the attribute names set up here exactly match the values you have in your account with the Identity Provider. These are the attributes the IdP will send in the response.

Step 5: Configure the Service Provider Account with the Identity Provider

Identity Provider user interfaces vary, but all essentially gather the same information that is needed for your SAML Web Browser SSO feature to work.

Provide the values you collected in Step 2: Gather Information on page 13.

Be ready with the metadata file generated by Policy Manager in Step 4: Configure the Domain in Policy Manager on page 14. Make sure you get the metadata.xml file for the container that has Community Manager installed (if the domain will be used for Community Manager login) and/or has the OAuth Provider feature installed (if the domain will be used for Community Manager OAuth domain, for resource owner authentication for at least one OAuth Provider).

If needed, you could refer to the two examples provided later in this publication:

- Identity Provider Configuration Example: SSO Circle on page 35
- Identity Provider Configuration Example: PingFederate on page 42

Step 6: Community Manager Configuration

This section includes procedures in Community Manager to complete the setup. It includes:

- Login Domain: see To enable a SAML login domain in Community Manager on page 18.
- Setting up the OAuth Provider Domain: see To configure a SAML OAuth Provider domain in Community Manager on page 18.

To enable a SAML login domain in Community Manager

1 In Community Manager, log in as the Site Admin.

2 Go to Administration > Config > Logins.

3 Find the domain on the list, and click **Enable**.

To configure a SAML OAuth Provider domain in Community Manager

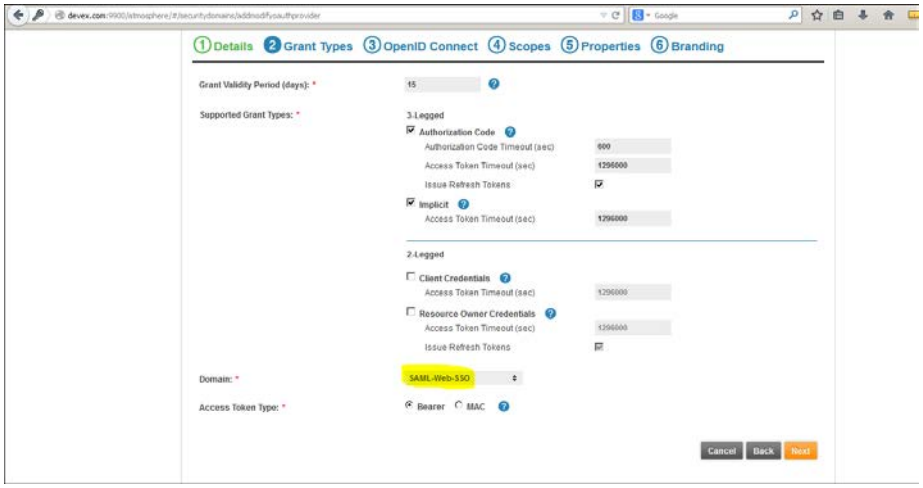
1 In Community Manager, log in as the Site Admin.

2 Go to Administration > Config > Domains.

3 Click **Add Domain** and choose **OAuth Provider**. The Add OAuth Provider wizard opens.

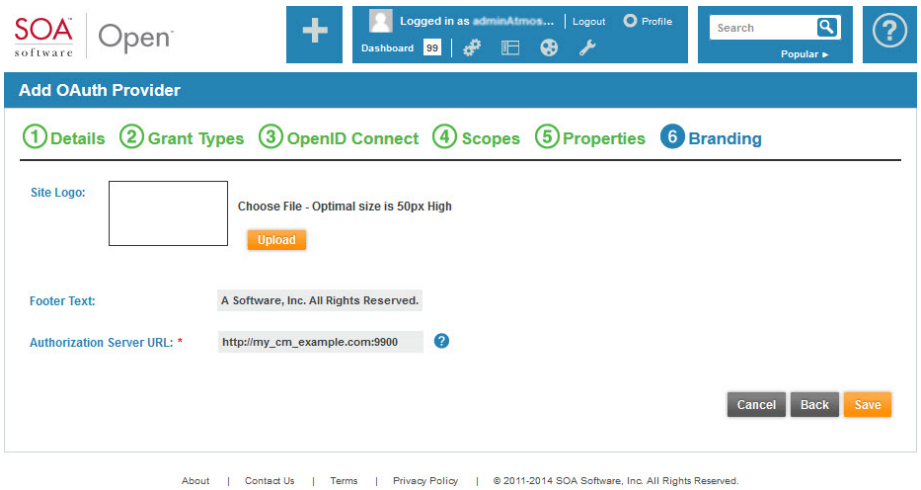
4 On the **Details** page, provide name and description, and click **Next**.

5 On the **Grant Types** page, select grant types and choose the SAML domain, as shown below, and then click **Next**.



6 At the **OpenID Connect** page, enable OpenID Connect, specify values as needed, and then click **Next**.

7 On the Branding page, provide the Authorization Server URL, as shown below, and click **Save**. The domain is created.



Note: When you create a new OAuth Provider Domain, you must also add the authorization server URL to your account with your SAML Identity Provider if it isn't already there. See Adding a New OAuth Provider Domain: Manual IdP Configuration on page 50.

Step 6: Test

Once you've completed the setup steps, it's important to test to make sure everything is working properly. Depending on how you're using the SAML Web Browser SSO functionality, complete either or both of the following:

- Testing the SAML Domain as a Login Domain on page 19
- Testing the SAML Domain as an OAuth Provider Domain on page 19

Testing the SAML Domain as a Login Domain

To test the login domain, follow the steps below.

To test the SAML login domain

1 Log out

2 Log in via your new login domain and verify that it works.

If you encounter any issues, check to make sure that the values in your IdP account and your SAML Web Browser SSO domain match.

You can also refer to the Troubleshooting section: see Troubleshooting on page 52.

Testing the SAML Domain as an OAuth Provider Domain

This section provides the steps you'll need to complete in Community Manager, after setting up a SAML domain as the OAuth Provider domain, to verify that the domain is correctly set up as the OAuth Provider domain and is working.

You'll need to complete a few steps to create a test scenario for the SAML domain. A high-level overview of the steps is given below. In some cases, a separate procedure is provided, in other cases it is not. If you need more information, refer to the Community Manager online help.

To test the SAML OAuth Provider domain: high-level procedure

1 Create an API that uses the SAML domain as the OAuth Provider.

See: Create API and specify OAuth Details on page 20

2 Create an app.

See: Create App on page 21.

3 Create a contract for the app with the API.

See: Request App/API Contract on page 21 and Approve App/API Contract on page 21.

4 Test the app in Dev Console to verify that the SSO Login screen is presented and the token is passed.

See: In Community Manager, Test in Dev Console on page 21.

If you encounter any issues, check to make sure that the values in your IdP account and your SAML Web Browser SSO domain match.

You can also refer to the Troubleshooting section: see Troubleshooting on page 52.

Create API and specify OAuth Details

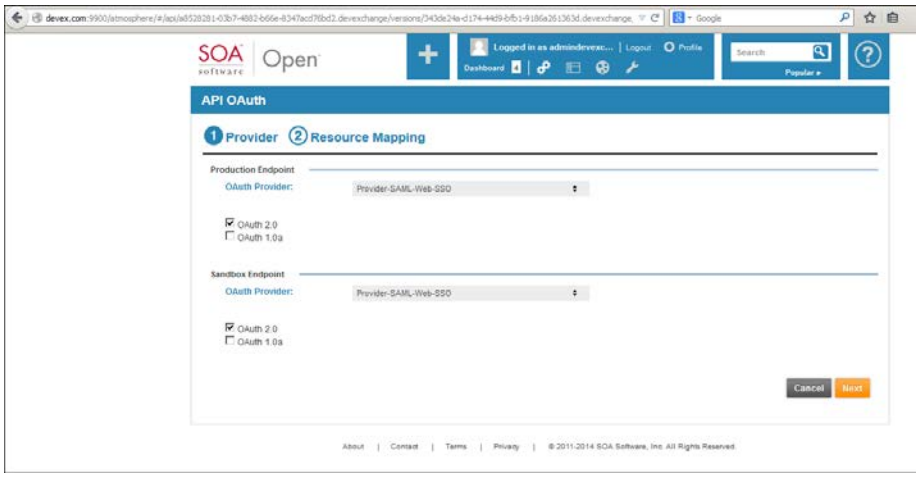
1 In Community Manager, select **Add a New API**. If necessary, choose **API with New Service**. The Add API Wizard displays at the first page.

2 Create the API, making sure you attach the following policies:

- OAuth Security Policy
- Detailed Auditing policy

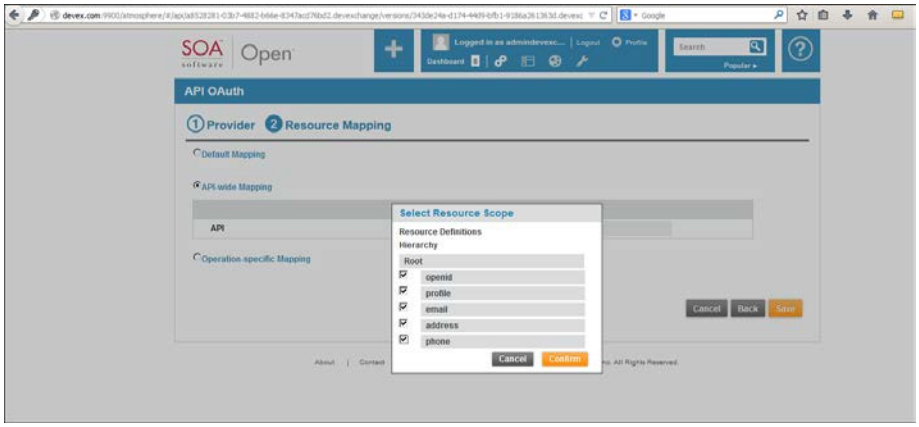
For more information, refer to the Community Manager online help.

3 Save the API and then, at the API Details page, click **OAuth Details** to access the API OAuth wizard, as shown below.



4 Select the SAML domain as the OAuth provider, and click **Next**.

5 In the **Resource Mapping** page, choose either API-wide Mapping or Operation-specific Mapping, specify scopes, and then click **Save**.



Create App

1 In Community Manager, select **Add a New App**.

2 Provide app details and then click **Save**.

Request App/API Contract

1 In Community Manager, search for the API you created, and click **Access**.

2 Choose the app you created, and complete the API Access wizard.

Approve App/API Contract

Unless you set up the API for auto-approval, you'll need to approve the API access request.

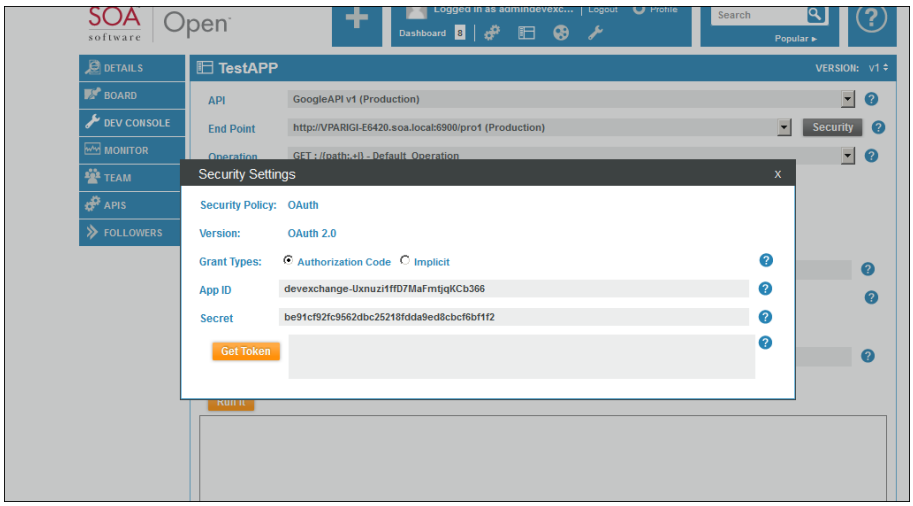
1 Go to your notifications and find the API access request you just made.

2 Approve the request.

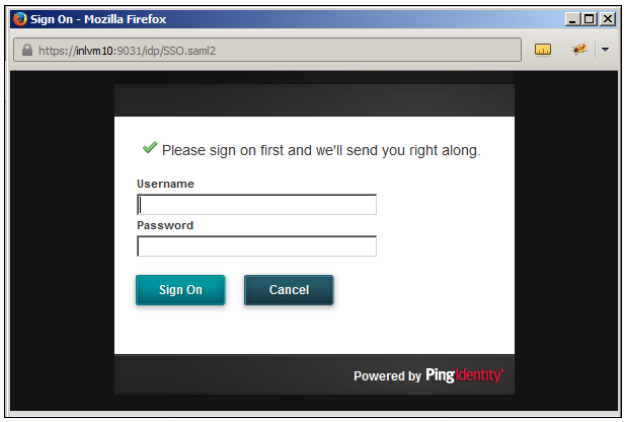
In Community Manager, Test in Dev Console

1 Go to the App Details page and click **Dev Console**.

2 Choose the API and click the **Security** button to view the security settings, as shown below.

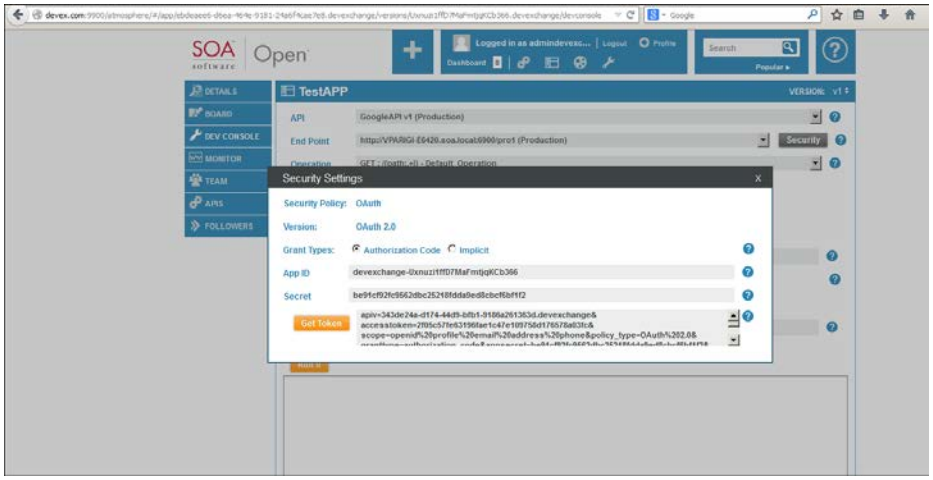


3 Choose the OAuth version and then click **Get Token**. The SSO login screen opens. An example is shown below.

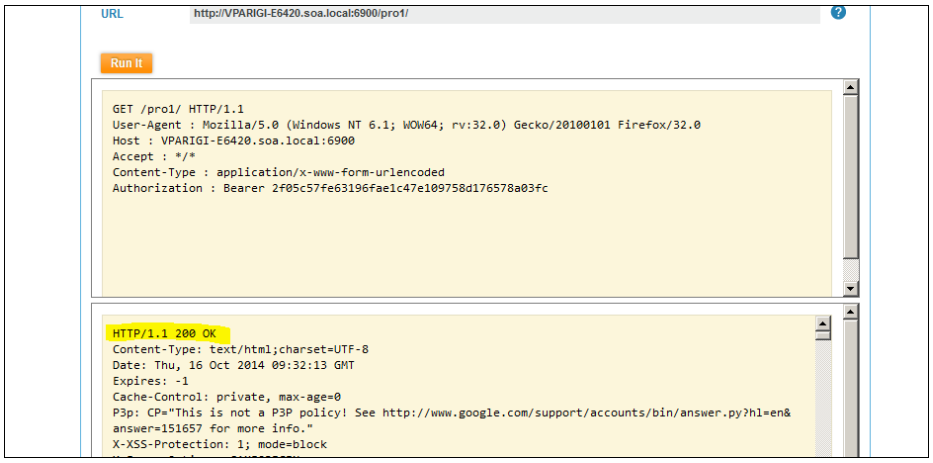


4 Provide username and password, and click Sign On. The Authorization window opens, as shown below.

5 Click **Authorize**. The token is retrieved, as shown below.



6 In Dev Console, click Run it. The API request is authorized, and runs successfully, as shown in the example below.



Chapter 4 | Sample Requests, Responses, and Metadata

This section includes some samples to show you what requests, responses, and metadata files might look like. It includes:

Sample Request: HTTP POST

The example below shows a sample HTTP POST request to SSO Circle.

Message Headers:

```
POST /sso/SSOPOST/metaAlias/ssocircle HTTP/1.1
Host: idp.ssocircle.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:32.0) Gecko/20100101 Firefox/32.0 openid
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://acmepaymentscorp.com/api/login/ssoLogin?domain=idp-
ssocircle&finalUrl=http%3A//acmepaymentscorp.com/ui/apps/atmosphere/_Vws1VQerwdBCGnF95K5OMUw/resourc
es/console/global/relyingpartypostlogin.html%3Fdynamic%3Dtrue%26baseUrl%3Dhttp%3A//acmepaymentscorp.com
/atmosphere
Cookie: JSESSIONID=F26FD035748B3706D17B6C850791FF7A; JROUTE=C9en;
__utma=161425727.1982119581.1415012088.1415012088.1415012088.1;
__utmz=161425727.1415012088.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none);
__utma=94376260.2017885730.1415012227.1415219015.1415528046.3;
__utmz=94376260.1415528046.3.3.utmccn=(referral)|utmcsr=acmepaymentscorp.com|utmcct=/api/login/ssoLogin|ut
mcmd=referral; amlbcookie=91; __utmc=94376260;
SSOCSession=AQIC5wM2LY4SfcxadFb3_TBcJQ6riqn7BuqUW0J6UEm01IA.*AAJTSQACMDIAAlNLABM1NTkzMT
EzNzk4NzA5NzQ1ODgwAAJTMQACMDE.*
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 4284
```

Message Body:

SAMLRequest=PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPHNhbWwycDpBdXRoblJlcXVlc3Q
gQXNzZXJ0aW9uQ29uc3VtZXJTZXJ2aWNlVVJMPSJodHRwOi8vYXRtb3NwaGVyZS5pbi9hcGkvbG9naW4vc3NvT
G9naW4iIERlc3RpbmF0aW9uPSJodHRwczovL2lkcC5zc29jaXJjbGUuY29tOjQ0My9zc28vU1NPUE9TVC9tZXRhQW
xpYXMvc3NvY2lyY2xlIiBGb3JjZUF1dGhuPSJmYWxzZSIgSUQ9Il84YmJkZTc2OTA4YzNiMmFhY2FkMDY1NmMzM
zFkNDgzMiIgSXNQYXNzaXZlPSJmYWxzZSIgSXNzdWVJbnN0YW50PSIyMDE0LTExLTA5VDEyOjMyOjM2LjQ0Mlo
iIFByb3RvY29sQmluZGluZz0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOmJpbmRpbmdzOkhUVFAtUE9TVCI
gVmVyc2lvbj0iMi4wIiB4bWxuczpzYW1sMnA9InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjIuMDpwcm90b2NvbCI%2B
PHNhbWwyOklzc3VlciB4bWxuczpzYW1sMj0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOmFzc2VydGlvbiI%2
Bc3Auc3NvY2lyY2xlLmNvbTwvc2FtbDI6SXNzdWVyPjxkczpTaWduYXR1cmUgeG1sbnM6ZHM9Imh0dHA6Ly93d3cu
dzMub3JnLzIwMDAvMDkveG1sZHNpZyMiPgo8ZHM6U2lnbmVkSW5mbz4KPGRzOkNhbm9uaWNhbGl6YXRpb25N
ZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxLzEwL3htbC1leGMtYzE0biMiLz4KPGRzOlNpZ
25hdHVyZU1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvMDkveG1sZHNpZyNyc2Etc2hhM
SIvPgo8ZHM6UmVmZXJlbmNlIFVSST0iI184YmJkZTc2OTA4YzNiMmFhY2FkMDY1NmMzMzFkNDgzMiI%2BCjxkcz
pUcmFuc2Zvcm1zPgo8ZHM6VHJhbnNmb3JtIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94b
Wxkc2lnI2VudmVsb3BlZC1zaWduYXR1cmUiLz4KPGRzOlRyYW5zZm9ybSBBbGdvcml0aG09Imh0dHA6Ly93d3cud
zMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPgo8L2RzOlRyYW5zZm9ybXM%2BCjxkczpEaWdlc3RNZXRob2
QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjc2hhMSIvPgo8ZHM6RGlnZXN0V
mFsdWU%2BRmkyMWZPZVBkQ3JHMnVqbkJjR2s0R1RLWUF3PTwvZHM6RGlnZXN0VmFsdWU%2BCjwvZHM6U
mVmZXJlbmNlPgo8L2RzOlNpZ25lZEluZm8%2BCjxkczpTaWduYXR1cmVWYWx1ZT4KWE9lSWUwTjFySHhyWTlrN
nZDWkYzVkM4eGV2T20rQXVpSktkOFkzUU84ZVc1U3NxdDJIazR3YllSeWJzNGpCK3MwSFdtSVpwTW9TNwpPbm
FpWWtSUVUxcW83S0x1QitvYWQ1dVJRYllqQ2x1NkswcjF2ZEFQdDhlN0lUYnF1TWFwbHZvMzI4RjB5cldRRVh0K0
pwVENyM1BpCms2TXpxTmpRRkJMY3BGakNJQzQ9CjwvZHM6U2lnbmF0dXJlVmFsdWU%2BCjxkczpLZXlJbmZvP
jxkczpYNTA5RGF0YT48ZHM6WDUwOUNlcnRpZmljYXRlPk1JSURIekNDQWdlZ0F3SUJBZ0lDQStzd0RRWUpLb1p
JaHZjTkFRRUZCUUF3Z2JneEN6QUpCZ05WQkFZVEFrbE9NUXN3Q1FZRFZRUUkKREFKVVV6RU1NQW9HQTF
VRUJ3d0RTRmxFTVRjd05RWURWUVFLREM1VFQwRWdVMjltZEhkaGNtVWdSVzVuYVc1bFpYSnBibWNnU1c1aw
phV0VnVUhKcGRtRjBaU0JNYVcxcGRHVmtNUlF3RWdZRFZRUUxEQXRGYm1kcGJtVmxjbWx1WnpFL01EMEdBM
VVFQXd3MlVHOXNhV041CklFMWhibUZuWlhJZ1EyVnlkR2xtYVdOaGRHVWdRWFYwYUc5eWFYUjVJQzBnUVVK
VlNrRkhUMDVFUVMwMk5ETXdNQjRYRFRFME1URXcKTlRJd01qQXlNVm9YRFRFMU1URXdOVEl3TWpBeU1Wb
3djREVMTUFrR0ExVUVCaE1DU1U0eEN6QUpCZ05WQkFnVEFsUlRNUXd3Q2dZRApWUVFIRXdOSVdVUXhGVE
FUQmdOVkJBb1RERk5QUVNCVGIyWjBkMkZ5WlRFVU1CSUdBMVVFQ3hNTFJXNW5hVzVsWlhKcGJtY3hHVEFY
CkJnTlZCQU1URUhOd0xuTnpiMk5wY21Oc1pTNWpiMjB3Z1o4d0RRWUpLb1pJaHZjTkFRRUJCUUFEZ1kwQU1JR
0pBb0dCQUpWKzJ4VE0KVXBaeWp3STdOTldoZS8zSndLNFF3cjhicGd5UkpDdjdWTC9qR2xEUTN0NGg1U0VYMlh
wMlpRQURVUUpuV3gwRmJ0ZFBZU29PdE1XLwoyNlFDY2YrSWkreit3dFcxYXlzTlJ0NEdqZkI0NXRmc0RTMWY3c
VpReTJ5MkVXSWRTN3FoRXlTa2JkWkpPR3l5cjNlbGpqdUx1Zm8rCjAzdVhYUkxqRitKRkFnTUJBQUV3RFFZSktvW
klodmNOQVFFRkJRQURnZ0VCQUJ3MjF2NTgyNTZhdDhtTFRVcExWMEU3MlZ4OFJ2b0YKbjY3OGZsN0ltcm56Qlh
6YmVraIRWR1Frc2E3SkNuZVpqM3NFTTM0L0lnZW9DL3dtRUVyazh5REh6UjRJR3ZUUXdwcUVpT01JUFg5cQpqQ
05zNHA0LzVndldvVU1nODdmM0dnb3JCQ1hYU3NZTUFCTzJHbGI1eSt4YkVtd2J5Ylhia1hIMVVsRDJ3a2pvN1VmQ
WVMVkRuT0RQCkpVSDRxc3hoakRvLzlJc0RKWk93ZnhmSXVhbHYvR2xzZkI3MzBvMkJibHVtYXp4UjE3bmh5Q2R
pYnhxalZhQnVYRUJyc2wrSzZLeHcKTWZ0Vk82QXpFcklxVVZVR3FEL0lkcUVhWXBOWEFEbkIzTkhNclE3Nmdnbm
srY2ZndDl4OXVHZzI4bi9WYm5HT1FUcEdLOVF0V1ZsVgpiZ3RBek1JPTwvZHM6WDUwOUNlcnRpZmljYXRlPjwvZ
HM6WDUwOURhdGE%2BPGRzOktleVZhbHVlPjxkczpSU0FLZXlWYWx1ZT48ZHM6TW9kdWx1cz5sWDdiRk14U2x
uS1BBanMwMWFGNy9jbkFyaERDdnh1bURKRWtLL3RVditNYVVORGUzaUhsSVJmWmVuWmxBQU5SQW1kYkhR
VnUxMDloCktnNjB4Yi9icEFKeC80aUw3UDdDMWJWckt3MUczZ2FOOEhqbTErd05MVi91cGxETGJMWVJZaDFMd
XFFVEpLUnQxa2s0YkxLdmQ2V08KTzR1NStqN1RlNWRkRXVNWDRrVT08L2RzOk1vZHVsdXM%2BPGRzOkV4cG
9uZW50PkFRQUI8L2RzOkV4cG9uZW50PjwvZHM6UlNBS2V5VmFsdWU%2BPC9kczpLZXlWYWx1ZT48L2RzOktle
UluZm8%2BPC9kczpTaWduYXR1cmU%2BPC9zYW1sMnA6QXV0aG5SZXF1ZXN0Pg%3D%3D&RelayState=Iw0KI
1N1biBOb3YgMDkgMTg6MDI6MzYgSVNUIDIwMTQNCmZpbmFsVXJsPWh0dHBcOi8vYXRtb3NwaGVyZS5pbi91aS
9hcHBzL2F0bW9zcGhlcmUvX1Z3czFWUWVyd2RCQ0duRjk1SzVPTVV3L3Jlc291cmNlcy9jb25zb2xlL2dsb2JhbC9y
ZWx5aW5ncGFydHlwb3N0bG9naW4uaHRtbD9keW5hbWljXD10cnVlJmJhc2VVcmxcPWh0dHBcOi8vYXRtb3NwaG
VyZS5pbi9hdG1vc3BoZXJlDQpzc29SZXRyeUNvdW50PTANCmRvbWFpbj1pZHAtc3NvY2lyY2xlDQo%3D

Sample Request: HTTP Redirect

The example below shows a sample HTTP Redirect request to SSO Circle.


```
GET
/sso/SSORedirect/metaAlias/ssocircle?SAMLRequest=nVbZkqrIFn33Kyo8j0YVg%2BAUp6ojGUQQUEbRlxsMCaJMk
iDo1zdqVZ06Fd19%2B943Mtl75dpr7dzw8482TZ7OsERxnr32iRe8%2FwQzPw%2FiLHrtW%2Bb8edL%2F4633E7lp
QhYzUFf7TIenGqLqCSAEy6rLY%2FMM1SksDVieYx9auvza31dVMcMwt0pzVOxhCV%2FiDHOLGEvyqHtCKJdvD%
2F0nroOKM7e6n3%2FLQl1aHBQvXYgfl34CX%2Fw8nVHU8JaEGcZKh0FcQr%2FCUli5IIldhH3G9p%2FmeenDO8%
2FXfugmqNsSudf%2Bf6Yjn4QT2p%2B4FOURFOlRY5%2BgCHI0hpTrDaddGFq7CMVn%2BCsRoRqKGarcrHrtkzhB
PRPEMz41CXJGTWdD%2BmU0nu76T%2Bsyr3I%2FT5g4ewhXl9ksd1GMZpmbQjSr%2FJkBFHlGvuAz7xGEZgvTX
D%2BDTsHQ9av%2Bk%2F1hAnkzobMlQ7OH7P%2BMV7wf3n97uDS7sy6%2FIvwzgPvhY%2F8NfZP9J%2FYV8u1
ngGZGHHVu1SV8PyBAn2Y3TfPSDF%2FyMsJIHMcxfIp1MQGKox%2F9rofek2EgZmH%2BWLNulmex7ybx9d4BCqz
2efAEkigv42qf%2Fg00gRH4DfoZtv5z52L2o4%2F9OuDO7l8i%2FUayRO4z2rvEB5gOw65xMx8%2BWbr42v%2Fx33
vokWeWbobCvEzRt%2FX%2FRgdmZ5jkBQye0UdVH8z%2BPeLfSIX9BU0ujrq7%2BP8I91W0B4rtJjV8O08nzCHT2il
c5M6I9y%2BHXWBwrLxolNc7ha%2FBD1Kfmr%2Bvv3fMp8HvSUfpsGFDa%2BGJwNw0mzaISrOogZSNDuuQrq3TZ
T5VXeo4rwdROjxf5RrT9tx8u%2BY35uI0dd0GL4VkSm1O3rWn6Ef%2BKkbWDoBIE%2F0dqNQNKubE2QQSw1iTUp
nw3GDQbvfXNeHEZJ3TC%2BvoLZG8ZQdlDOlmDXXULMPwLPXCjMhdwtdHcjkYDoajBca%2F%2Fqrpaw23upbwci
%2Fy9uzQ%2BJRzK%2Fdzwd7uZ9jdkwq%2BKaLILa4sCyIYgUZkQCSyYIAaTttKy3wn7s%2B%2BCjR%2Bzmigibyo
Za8d90i1GbA1wTFZKRpqWG3L2Zom9jggWdaVVxSQC4CweKZpOGOe8IrpN%2BojaMmxtGnifGSR0zRYBHs%2Ft
SJ9Q2fuhk52jlR4qR8ZG%2FrYczdd0EIqgnSO7wxGcTdEEQj2UdG1ho%2FuYDIHqrmXBl2SnfhpUu%2BuPKZw%2B
P1w0DSkJUyRu1HpnsgTey%2BdZztHjDTSvgRCknYv9oFgRZpj464wvbiOTossHmmWZBvH%2BcLEaV4zcFLlIEZhK
IczeVwx%2BaanmmKjHMBFsfPbJnHb%2FLLX%2BBwvK%2BD4kGCvsIZFfdEtMkGim4rWNGxXRa8rY9GlixtLa%2B
cmMB9BucnN1bVmMKZH7vCAnF92Jm8pjHgHZVtF%2FqZZK5jA6T2SOz4LtZEz9eqRauGnKtoZ9MEj8SbaTb47y3T
OctEWB4ooSJ1vDJDsAdmaSs8qdpdDI45VdbOH2FBqlpTWlBOviC66xJ7HtowdhITThhW1pw3eIZ2C3HVgliZlmxafe1
Ww3hr5qlI2WI8caawfDsR4cB001YZwL0jVK0o4hAxFVyHiDCIcn3bahbyQ%2FEYMjPFpz1%2BMoxfspJVwuZRDmB
wOtVyH%2BaCHD2vH0eXDfCDNQaQwAPB%2F1bBcFPEMYBqSONMTkh651SSVTauQbZwfk3Y70c%2F5vJeNxp
MwGYtpmV0Z5%2BrB48G0Be2I3LHEZnB3GCJeGVKYGMGcxZqU58vj5NJdGp0ShbOpNcWJj1eKuHamp96BVRFV
UBgdnTe5pUSTcTgUutHHsI5joK0CmBUpJB59GbQenzbexXO8o7MgrIQjm%2BMhH1shgLLNZStu3ZOsBXVC7f7A5
dhURJy0WzVhG4q1m5wxIUEhMx7iOcl4SZ2611Ynxtn%2Bwgax154OtsvUDs%2BUKBksR8u26SlhZa9G4MqX4smy
LeHEYWJw4t1toTqAy5ihulBKbTyKouw48MOomrbTWojISYbZXiasNLMQllOt2tiJ3fOiClwV8TF1v4%2BUz83PodNNo8
dsun8GDfDbWsmDOqnRW%2BKMvbnSGkm2XIMDwgl3Psb8DJR7jj23dcpJ%2FHGJVdZ5oLiWysFhvEhEPdzBbJcA
oOogDbyFZtcEPt33ltEIbz3MK4DUYlQsj9djlvDsctkQwjBy1cnikBKDRpVtrC4STvbkrb7dE3J94k1pqVfE8Uh58vIcjDar3
oqq6cFhbEI6CPhacaij9aj8g%2FqtDL4t8gxm1RvQAHN%2F%2B7nz%2BBJ9LRv7TRTs68D%2BbaK%2Fvf8x%2Ff6r
%2FPYn&RelayState=Iw0KI1N1biBOb3YgMDkgMTg6MTk6MzUgSVNUIDIwMTQNCmZpbmFsVXJsPWh0dHBcOi8v
YXRtb3NwaGVyZS5pbi91aS9hcHBzL2F0bW9zcGhlcmUvX1Z3czFWUWVyd2RCQ0duRjk1SzVPTVV3L3Jlc291cmNl
cy9jb25zb2xlL2dsb2JhbC9yZWx5aW5ncGFydHlwb3N0bG9naW4uaHRtbD9keW5hbWljXD10cnVlJmJhc2VVcmxcP
Wh0dHBcOi8vYXRtb3NwaGVyZS5pbi9hdG1vc3BoZXJlDQpzc29SZXRyeUNvdW50PTANCmRvbWFpbj1pZHAtc3Nv
Y2lyY2xlDQo=&SigAlg=http%3A%2F%2Fwww.w3.org%2F2000%2F09%2Fxmldsig%23rsa-
sha1&Signature=QAJ6BU5zuIytWuTEdwqw%2BI9uSm3QVgo9n1REX7FqLkXo%2BpH%2BDIdy4XheIkytyaDfnnDcV
cOH7vQCSw4DbzdH30Unmg%2BEqjbUTbcPvgyJvv%2FKKBS%2FxyYDKNDyqTyUYq1ao%2Fspa3rtmZixki00VuU
Yo7PrZzjrGjHMyQ6ycfxsMDo%3D HTTP/1.1
Host: idp.ssocircle.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0 openid
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://acmepaymentscorp.com/atmosphere/
Cookie: JSESSIONID=F26FD035748B3706D17B6C850791FF7A; JROUTE=C9en;
__utma=161425727.1982119581.1415012088.1415012088.1415012088.1;
__utmz=161425727.1415012088.1.1.utmcsr=(direct)|utmccn=(direct)|utmcmd=(none);
__utma=94376260.2017885730.1415012227.1415219015.1415528046.3;
__utmz=94376260.1415528046.3.3.utmccn=(referral)|utmcsr=acmepaymentscorp.com|utmcct=/api/login/ssoLogin|ut
mcmd=referral; amlbcookie=91;
SSOCSession=AQIC5wM2LY4SfcxadFb3_TBcJQ6riqn7BuqUW0J6UEm01IA.*AAJTSQACMDIAAlNLABM1NTkzMT
EzNzk4NzA5NzQ1ODgwAAJTMQACMDE.*
Connection: keep-alive
```

Sample Response: HTTP POST

The example below shows a sample HTTP POST response from SSO Circle.

```
POST /api/login/ssoLogin HTTP/1.1
Host: acmepaymentscorp.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:32.0) Gecko/20100101 Firefox/32.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: JSESSIONID_platform=1h2mr9mg0ik1n8dxmliqmh2tf
Content-Type: application/x-www-form-urlencoded
Content-Length: 6741
```

Message Body:


```
SAMLResponse=PHNhbWxwOlJlc3BvbnNlIHhtbG5zOnNhbWxwPSJ1cm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLj
A6%0D%0AcHJvdG9jb2wiIElEPSJzMmZkMjEzNjIyZWIyOTI0ZDdmYTg1ODg2MjYzNzVmOTM1MDJjMDFhZTAi%0
D%0AIEluUmVzcG9uc2VUbz0iXzhiYmRlNzY5MDhjM2IyYWFjYWQwNjU2YzMzMWQ0ODMyIiBWZXJzaW9u%0D%
0APSIyLjAiIElzc3VlSW5zdGFudD0iMjAxNC0xMS0wOVQxMjozMjozN1oiIERlc3RpbmF0aW9uPSJo%0D%0AdHRwO
i8vYXRtb3NwaGVyZS5pbi9hcGkvbG9naW4vc3NvTG9naW4iPjxzYW1sOklzc3VlciB4bWxu%0D%0AczpzYW1sPSJ1
cm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6YXNzZXJ0aW9uIj5odHRwOi8vaWRw%0D%0ALnNzb2NpcmNsZS
5jb208L3NhbWw6SXNzdWVyPjxzYW1scDpTdGF0dXMgeG1sbnM6c2FtbHA9InVy%0D%0AbjpvYXNpczpuYW1lczp
0YzpTQU1MOjIuMDpwcm90b2NvbCI%2BCjxzYW1scDpTdGF0dXNDb2RlICB4%0D%0AbWxuczpzYW1scD0idXJuO
m9hc2lzOm5hbWVzOnRjOlNBTUw6Mi4wOnByb3RvY29sIgpWYWx1ZT0i%0D%0AdXJuOm9hc2lzOm5hbWVzOnRj
OlNBTUw6Mi4wOnN0YXR1czpTdWNjZXNzIj4KPC9zYW1scDpTdGF0%0D%0AdXNDb2RlPgo8L3NhbWxwOlN0YX
R1cz48c2FtbDpBc3NlcnRpb24geG1sbnM6c2FtbD0idXJuOm9h%0D%0Ac2lzOm5hbWVzOnRjOlNBTUw6Mi4wOmF
zc2VydGlvbiIgSUQ9InMyMTJhNTI3ODJiYjM1Y2E0NTZk%0D%0ANGFmNzViYTc3NTkzZGMwZTc0ZTA0YSIgSXNz
dWVJbnN0YW50PSIyMDE0LTExLTA5VDEyOjMyOjM3%0D%0AWiIgVmVyc2lvbj0iMi4wIj4KPHNhbWw6SXNzdWVy
Pmh0dHA6Ly9pZHAuc3NvY2lyY2xlLmNvbTwv%0D%0Ac2FtbDpJc3N1ZXI%2BPGRzOlNpZ25hdHVyZSB4bWxuczp
kcz0iaHR0cDovL3d3dy53My5vcmcvMjAw%0D%0AMC8wOS94bWxkc2lnIyI%2BCjxkczpTaWduZWRJbmZvPgo8ZH
M6Q2Fub25pY2FsaXphdGlvbk1ldGhv%0D%0AZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMTA
veG1sLWV4Yy1jMTRuIyIvPgo8%0D%0AZHM6U2lnbmF0dXJlTWV0aG9kIEFsZ29yaXRobT0iaHR0cDovL3d3dy53M
y5vcmcvMjAwMC8wOS94%0D%0AbWxkc2lnI3JzYS1zaGExIi8%2BCjxkczpSZWZlcmVuY2UgVVJJPSIjczIxMmE1Mj
c4MmJiMzVjYTQ1%0D%0ANmQ0YWY3NWJhNzc1OTNkYzBlNzRlMDRhIj4KPGRzOlRyYW5zZm9ybXM%2BCjxkcz
pUcmFuc2Zvcm0g%0D%0AQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjZW52Z
WxvcGVkLXNp%0D%0AZ25hdHVyZSIvPgo8ZHM6VHJhbnNmb3JtIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcm
cvMjAw%0D%0AMS8xMC94bWwtZXhjLWMxNG4jIi8%2BCjwvZHM6VHJhbnNmb3Jtcz4KPGRzOkRpZ2VzdE1ldGhv
ZCBB%0D%0AbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDAvMDkveG1sZHNpZyNzaGExIi8%2BCjxkczp
E%0D%0AaWdlc3RWYWx1ZT5ZVTY5V0ppNjZudDROL2dkMExJL016TlVRUUk9PC9kczpEaWdlc3RWYWx1ZT4K
%0D%0APC9kczpSZWZlcmVuY2U%2BCjwvZHM6U2lnbmVkSW5mbz4KPGRzOlNpZ25hdHVyZVZhbHVlPgphbUhN
%0D%0AMjg1ZTcrWkxwOXZxejdabkk0WGV4L2EybU1tcE55YVdwRGpQbGFXY3N0OGhOdVRPVlVGd3hxQ2NO%
0D%0AV0xWTzBCVHJKZTVpelhOCndIT3NmWS9DT1RqWTlqcnZPQ3hSR0pxakg0Z1pqU3p5clBYYlVPNGdu%0D
%0AcDhoanlvbm80TG5QTm1rR0l0Nk1QU1pGNnY0NE1qUE05N3AKZW14TldlZGtaeFVFTGZ4KzRSTT0K%0D%0
APC9kczpTaWduYXR1cmVWYWx1ZT4KPGRzOktleUluZm8%2BCjxkczpYNTA5RGF0YT4KPGRzOlg1MDlD%0D%0
AZXJ0aWZpY2F0ZT4KTUlJQ2pEQ0NBWFNnQXdJQkFnSUZBSlJ2eGNNd0RRWUpLb1pJaHZjTkFRRUVC%0D%0
AUUF3TGpFTE1Ba0dBMVVFQmhNQ1JFVXhFakFRQmdOVgpCQW9UQ1ZOVFQwTnBjbU5zWlRFTE1Ba0dB%0D
%0AMVVFQXhNQ1EwRXdIaGNOTVRFd05URTNNVGsxTnpJeFdoY05NVFl3T0RFM01UazFOekl4CldqQkxN%0D%
0AUXN3Q1FZRFZRUUdFd0pFUlRFU01CQUdBMVVFQ2hNSlUxTlBRMmx5WTJ4bE1Rd3dDZ1lEVlFRTEV3%0D%0
ATnBaSEF4R2pBWUJnTlYKQkFNVEVXbGtjQzV6YzI5amFYSmpiR1V1WTI5dE1JR2ZNQTBHQ1NxR1NJ%0D%0A
YjNEUUVCQVFVQUE0R05BRENCaVFLQmdRQ2J6RFJrdWRDLwphQzJnTXFSVlZhTGRQSkpFd3BGQjRv%0D%0
ANzFmUjVibk5kMm9jbm5OekovVzlDb0Nhcmd6S3grRUo0Tm0zdldtWC9JWlJDRnZydnk5Qzc4CmZQ%0D%0AMW
NtdDZTYTA5MUs5bHVhTUF5V243b0M4aC9ZQlhIN3JCNDJ0ZHZXTFk0S2w5Vkp5NlVDY2x2YXN5%0D%0AcmZLe
CtTUjRLVTZ6Q3NNNjIKMkt2cDV3VzY3UUlEQVFBQm94Z3dGakFVQmdsZ2hrZ0JodmhDQVFF%0D%0AQkFmOEV
CQU1DQkhBd0RRWUpLb1pJaHZjTkFRRUVCUUFEZ2dFQgpBSjBoZXVhN21GTzNRc3pkR3Ux%0D%0ATmJsR2F
URFh0ZjZUeHRlMHpwWUl0KzhZVWN6YTJTYVpYWHZDTGI5RHZHeFcxVEpXYVpwUEdwSHo1%0D%0ACnRMW
EpiZFlRbjd4VEFuTDR5UU9LTjZ1TnFVQS9hVFZneXlVSmtXWnQyZ2l3RXNXVXZHMFVCTVNQ%0D%0AUzF0cDJ
wVjJjNi9vbEljYmRZVTYKWmVjVXo2TjI0c1NTN2l0RUJDNm53Q1ZCb0hPTDh1Nk1zZnhN%0D%0ATER6SklQQkk2
```

OFVaanozSU1LVERVRHY2VTlEdFltWExjOGlNVlpCbgpjWUpuOU5nTmkzZ2hsOWZZ%0D%0AUHBIY2M2UWJYZU
RVamhkelhYVXFHK2hCNkZhYkdxZFRka0lad29pNGdOcHlyM2thY0tSVldKc3NE%0D%0AZ2FrCmVMMk1vRE5xSnl
RMGZYQzZaZTNmNzlDS3kvV2plVTVGTHdEWIIwUT0KPC9kczpYNTA5Q2Vy%0D%0AdGlmaWNhdGU%2BCjwvZ
HM6WDUwOURhdGE%2BCjwvZHM6S2V5SW5mbz4KPC9kczpTaWduYXR1cmU%2BPHNh%0D%0AbWw6U3Via
mVjdD4KPHNhbWw6TmFtZUlEIEZvcm1hdD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6%0D%0AMi4wOm5hbWV
pZC1mb3JtYXQ6cGVyc2lzdGVudCIgTmFtZVF1YWxpZmllcj0iaHR0cDovL2lkcC5z%0D%0Ac29jaXJjbGUuY29tIj5lTV
VKK01wbENKRjYwSXBXUG5tUnhmbjN3bUhuPC9zYW1sOk5hbWVJRD48%0D%0Ac2FtbDpTdWJqZWN0Q29uZml
ybWF0aW9uIE1ldGhvZD0idXJuOm9hc2lzOm5hbWVzOnRjOlNBTUw6%0D%0AMi4wOmNtOmJlYXJlciI%2BCjxzYW
1sOlN1YmplY3RDb25maXJtYXRpb25EYXRhIEluUmVzcG9uc2VU%0D%0Abz0iXzhiYmRlNzY5MDhjM2IyYWFjYWQ
wNjU2YzMzMWQ0ODMyIiBOb3RPbk9yQWZ0ZXI9IjIwMTQt%0D%0AMTEtMDlUMTI6NDI6MzdaIiBSZWNpcGllbnQ9
Imh0dHA6Ly9hdG1vc3BoZXJlLmluL2FwaS9sb2dp%0D%0Abi9zc29Mb2dpbiIvPjwvc2FtbDpTdWJqZWN0Q29uZmly
bWF0aW9uPgo8L3NhbWw6U3ViamVjdD48%0D%0Ac2FtbDpDb25kaXRpb25zIE5vdEJlZm9yZT0iMjAxNC0xMS0wO
VQxMjoyMjozN1oiIE5vdE9uT3JB%0D%0AZnRlcj0iMjAxNC0xMS0wOVQxMjo0MjozN1oiPgo8c2FtbDpBdWRpZW5jZ
VJlc3RyaWN0aW9uPgo8%0D%0Ac2FtbDpBdWRpZW5jZT5zcC5zc29jaXJjbGUuY29tPC9zYW1sOkF1ZGllbmNlPgo
8L3NhbWw6QXVk%0D%0AaWVuY2VSZXN0cmljdGlvbj4KPC9zYW1sOkNvbmRpdGlvbnM%2BCjxzYW1sOkF1dGh
uU3RhdGVtZW50%0D%0AIEF1dGhuSW5zdGFudD0iMjAxNC0xMS0wOVQxMDoxNDoxM1oiIFNlc3Npb25JbmRleD
0iczI5NDhi%0D%0AMGU1NjY0YmVlNTJmMzE0NmI4ZDg1YjM0ZmUyOGEzYTc2MzAxIj48c2FtbDpBdXRobkNvbnR
leHQ%2B%0D%0APHNhbWw6QXV0aG5Db250ZXh0Q2xhc3NSZWY%2BdXJuOm9hc2lzOm5hbWVzOnRjOlNBTU
w6Mi4wOmFj%0D%0AOmNsYXNzZXM6UGFzc3dvcmRQcm90ZWN0ZWRUcmFuc3BvcnQ8L3NhbWw6QXV0aG5D
b250ZXh0Q2xh%0D%0Ac3NSZWY%2BPC9zYW1sOkF1dGhuQ29udGV4dD48L3NhbWw6QXV0aG5TdGF0ZW1lbn
Q%2BPHNhbWw6QXR0%0D%0AcmlidXRlU3RhdGVtZW50PjxzYW1sOkF0dHJpYnV0ZSBOYW1lPSJFbWFpbEFkZ
HJlc3MiPjxzYW1s%0D%0AOkF0dHJpYnV0ZVZhbHVlIHhtbG5zOnhzPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxL1h
NTFNjaGVt%0D%0AYSIgeG1sbnM6eHNpPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxL1hNTFNjaGVtYS1pbnN0YW5j
ZSIg%0D%0AeHNpOnR5cGU9InhzOnN0cmluZyI%2BYmFuaWwuY2JpdEBnbWFpbC5jb208L3NhbWw6QXR0cmlid
XRl%0D%0AVmFsdWU%2BPC9zYW1sOkF0dHJpYnV0ZT48c2FtbDpBdHRyaWJ1dGUgTmFtZT0iRmlyc3ROYW1lIj
48%0D%0Ac2FtbDpBdHRyaWJ1dGVWYWx1ZSB4bWxuczp4cz0iaHR0cDovL3d3dy53My5vcmcvMjAwMS9YTUxT%
0D%0AY2hlbWEiIHhtbG5zOnhzaT0iaHR0cDovL3d3dy53My5vcmcvMjAwMS9YTUxTY2hlbWEtaW5zdGFu%0D%0A
Y2UiIHhzaTp0eXBlPSJ4czpzdHJpbmciPkFuaWw8L3NhbWw6QXR0cmlidXRlVmFsdWU%2BPC9zYW1s%0D%0AO
kF0dHJpYnV0ZT48c2FtbDpBdHRyaWJ1dGUgTmFtZT0iTGFzdE5hbWUiPjxzYW1sOkF0dHJpYnV0%0D%0AZVZhb
HVlIHhtbG5zOnhzPSJodHRwOi8vd3d3LnczLm9yZy8yMDAxL1hNTFNjaGVtYSIgeG1sbnM6%0D%0AeHNpPSJodH
RwOi8vd3d3LnczLm9yZy8yMDAxL1hNTFNjaGVtYS1pbnN0YW5jZSIgeHNpOnR5cGU9%0D%0AInhzOnN0cmluZyI
%2BQnVqYWdvbmRhPC9zYW1sOkF0dHJpYnV0ZVZhbHVlPjwvc2FtbDpBdHRyaWJ1%0D%0AdGU%2BPC9zYW1
sOkF0dHJpYnV0ZVN0YXRlbWVudD48L3NhbWw6QXNzZXJ0aW9uPjwvc2FtbHA6UmVz%0D%0AcG9uc2U%2B&R
elayState=Iw0KI1N1biBOb3YgMDkgMTg6MDI6MzYgSVNUIDIwMTQNCmZpbmFsVXJsPWh0dHBcOi8vYXRtb3Nwa
GVyZS5pbi91aS9hcHBzL2F0bW9zcGhlcmUvX1Z3czFWUWVyd2RCQ0duRjk1SzVPTVV3L3Jlc291cmNlcy9jb25zb2
xlL2dsb2JhbC9yZWx5aW5ncGFydHlwb3N0bG9naW4uaHRtbD9keW5hbWljXD10cnVlJmJhc2VVcmxcPWh0dHBcOi
8vYXRtb3NwaGVyZS5pbi9hdG1vc3BoZXJlDQpzc29SZXRyeUNvdW50PTANCmRvbWFpbj1pZHAtc3NvY2lyY2xlDQ
o%3D

Sample Response: HTTP Artifact

The example below shows a sample HTTP Artifact response from SSO Circle.

GET
/api/login/ssoLogin?SAMLart=AAQAALN%2Bk3vq4G80Xko1XPLwwxsvPbU%2F0k5pJmYcpWTJarjtzdkp9Q2yMDE
%3D&RelayState=Iw0KI1N1biBOb3YgMDkgMTg6NDA6MzEgSVNUIDIwMTQNCmZpbmFsVXJsPWh0dHBcOi8vYX
Rtb3NwaGVyZS5pbi91aS9hcHBzL2F0bW9zcGhlcmUvX1Z3czFWUWVyd2RCQ0duRjk1SzVPTVV3L3Jlc291cmNlcy
9jb25zb2xlL2dsb2JhbC9yZWx5aW5ncGFydHlwb3N0bG9naW4uaHRtbD9keW5hbWljXD10cnVlJmJhc2VVcmxcPWh
0dHBcOi8vYXRtb3NwaGVyZS5pbi9hdG1vc3BoZXJlDQpzc29SZXRyeUNvdW50PTANCmRvbWFpbj1pZHAtc3NvY2l
yY2xlDQo%3D HTTP/1.1
Host: acmepaymentscorp.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:33.0) Gecko/20100101 Firefox/33.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://acmepaymentscorp.com/atmosphere/
Cookie: JSESSIONID_platform=1fjtfzek7t4vk2w6thha0481g;

Sample Metadata File: Identity Provider

The example below is the generic IdP metadata file published by SSO circle at http://idp.ssocircle.com/.


```
<EntityDescriptor entityID="http://idp.ssocircle.com">
  <IDPSSODescriptor WantAuthnRequestsSigned="false"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <KeyDescriptor use="signing">
      <ds:KeyInfo>
        <ds:X509Data>
          <ds:X509Certificate>X_509_certificate_value</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </KeyDescriptor>
    <KeyDescriptor use="encryption">
      <ds:KeyInfo>
        <ds:X509Data>
        <ds:X509Certificate>X_509_certificate_value</ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
      <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc">
        <xenc:KeySize>128</xenc:KeySize>
      </EncryptionMethod>
    </KeyDescriptor>
    <ArtifactResolutionService index="0" isDefault="true" Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="https://idp.ssocircle.com:443/sso/ArtifactResolver/metaAlias/ssocircle"/>
    <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://idp.ssocircle.com:443/sso/IDPSloRedirect/metaAlias/ssocircle"
ResponseLocation="https://idp.ssocircle.com:443/sso/IDPSloRedirect/metaAlias/ssocircle"/>
    <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://idp.ssocircle.com:443/sso/IDPSloPost/metaAlias/ssocircle"
ResponseLocation="https://idp.ssocircle.com:443/sso/IDPSloPost/metaAlias/ssocircle"/>
    <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="https://idp.ssocircle.com:443/sso/IDPSloSoap/metaAlias/ssocircle"/>
    <ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://idp.ssocircle.com:443/sso/IDPMniRedirect/metaAlias/ssocircle"
ResponseLocation="https://idp.ssocircle.com:443/sso/IDPMniRedirect/metaAlias/ssocircle"/>
    <ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://idp.ssocircle.com:443/sso/IDPMniPOSTmetaAlias/ssocircle"
ResponseLocation="https://idp.ssocircle.com:443/sso/IDPMniPOST/metaAlias/ssocircle"/>
    <ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="https://idp.ssocircle.com:443/sso/IDPMniSoap/metaAlias/ssocircle"/>
    <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent</NameIDFormat>
    <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>
    <NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</NameIDFormat>
    <NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</NameIDFormat>
    <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos</NameIDFormat>
    <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://idp.ssocircle.com:443/sso/SSORedirect/metaAlias/ssocircle"/>
    <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://idp.ssocircle.com:443/sso/SSOPOST/metaAlias/ssocircle"/>
    <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="https://idp.ssocircle.com:443/sso/SSOSoap/metaAlias/ssocircle"/>
    <NameIDMappingService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="https://idp.ssocircle.com:443/sso/NIMSoap/metaAlias/ssocircle"/>
  </IDPSSODescriptor>
</EntityDescriptor>
```

Sample Metadata File: Service Provider

In the sample metadata file shown below, the Service Provider is using SSO Circle as the Identity Provider.

For the sake of readability, certificates have been removed from the example below, and have been replaced with placeholders.


```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor ID="sp.ssocircle.com" entityID="sp.ssocircle.com"
xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
  <md:SPSSODescriptor AuthnRequestsSigned="true" WantAssertionsSigned="true"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>{x.509_Certificate_Goes_here}</ds:X509Certificate>
        </ds:X509Data>
        <ds:KeyValue>
          <ds:RSAKeyValue>

<ds:Modulus>lX7bFMxSlnKPAjs01aF7/cnArhDCvxumDJEkK/tUv+MaUNDe3iHlIRfZenZlAANRAmdbHQVu109h
Kg60xb/bpAJx/4iL7P7C1bVrKw1G3gaN8Hjm1+wNLV/uplDLbLYRYh1LuqETJKRt1kk4bLKvd6WO
O4u5+j7Te5ddEuMX4kU=</ds:Modulus>
            <ds:Exponent>AQAB</ds:Exponent>
          </ds:RSAKeyValue>
        </ds:KeyValue>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:KeyDescriptor use="encryption">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>{x.509_Certificate_Goes_here}</ds:X509Certificate>
        </ds:X509Data>
        <ds:KeyValue>
          <ds:RSAKeyValue>

<ds:Modulus>lX7bFMxSlnKPAjs01aF7/cnArhDCvxumDJEkK/tUv+MaUNDe3iHlIRfZenZlAANRAmdbHQVu109h
Kg60xb/bpAJx/4iL7P7C1bVrKw1G3gaN8Hjm1+wNLV/uplDLbLYRYh1LuqETJKRt1kk4bLKvd6WO
O4u5+j7Te5ddEuMX4kU=</ds:Modulus>
            <ds:Exponent>AQAB</ds:Exponent>
          </ds:RSAKeyValue>
        </ds:KeyValue>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent</md:NameIDFormat>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</md:NameIDFormat>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</md:NameIDFormat>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</md:NameIDFormat>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:kerberos</md:NameIDFormat>
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="http://acmepaymentscorp.in/saml/ACS/default" index="0" isDefault="true" />
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="http://acmepaymentscorp.in/api/login/ssoLogin" index="1" />
  </md:SPSSODescriptor>
</md:EntityDescriptor>
```

Sample Artifact Resolve Request

The example below shows a sample artifact resolve request where HTTP Artifact is used as the binding for the SAML response.


```
<saml2p:ArtifactResolve Destination="https://inlvm10:9031/idp/ARS.ssaml2"
ID="_44213af2e2143e460bbaab99c5f3d76c" IssueInstant="2014-10-20T09:55:15.783Z" Version="2.0"
xmlns:saml2p="urn:oasis:names:tc:SAML:2.0:protocol" xmlns:soap11="http://schemas.xmlsoap.org/soap/envelope/">
  <saml2:Issuer xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">sp.redirect.in</saml2:Issuer>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="#_44213af2e2143e460bbaab99c5f3d76c">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>YoYOhIrJ9sHFjdp88KsX2tLdwKc=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>

<ds:SignatureValue>BZP86nT4Zlo0X9XAsA0TnGNLOWb+Bozoo351lsxK3KWb8Jd1OnrZ+x0dMQJwS+3NjCJzvP/3
PYve
NXCv+qoM9SGM0mYj/AVNB9G4ssqiONT6GBp3S2QH47mzU68OS9S0uXEdbIJAoU7SSdRuNWX/o01H
C1pk25fPUTssLry28Jk=</ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>

<ds:X509Certificate>MIICPzCCAaigAwIBAgIIT3bnFBcGuFIwDQYJKoZIhvcNAQEFBQAwYjELMAkGA1UEBhMCSU
4xCzAJ
BgNVBAgTAlRTMQwwCgYDVQQHEwNIWUQxDDAKBgNVBAoTA1NPQTERMA8GA1UECxMIUHJvZHVjdHMx
FzAVBgNVBAMTDnNwLnJlZGlyZWN0LmluMB4XDTE0MTAyMDA5NDU1OFoXDTE1MTAyMDA5NDU1OFow
YjELMAkGA1UEBhMCSU4xCzAJBgNVBAgTAlRTMQwwCgYDVQQHEwNIWUQxDDAKBgNVBAoTA1NPQTER
MA8GA1UECxMIUHJvZHVjdHMxFzAVBgNVBAMTDnNwLnJlZGlyZWN0LmluMIGfMA0GCSqGSIb3DQEB
AQUAA4GNADCBiQKBgQCrRsJmI1eRHRcMwjHUxytdC3wp79yKOg0U3Zx9bC3N6kSXPcYOlnd+KjIs
ChRG1mYldW1ahvmByGKM4apII0Y2q3N2j91cDwJeGFd9b9tMnJHTWSDH8b1rAbF2zCQ45TDmJar+
FZefSzvtc3tOkt11Fc/AGhVOEsHDhP5p/QiySwIDAQABMA0GCSqGSIb3DQEBBQUAA4GBAHUVlMhh
qYdT9gxqSRBE2ZdzCgCKdtT5QgihHoPH6Zsl/52OkIcOUIyHO5qZ1eXW9VsD79kmBtP6fYCJ07G3
hO7AzWRsEa+wp/Nts6D91IO+MKocGdMC7m8I1cY8ZmArbExK0NZa40Kl0/oXZbDem6td3+9udLt9
nQ3QR27abcti</ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>

<saml2p:Artifact>AAQAANMHz4xHH5RgozwuezNtu6pBYWxe3CLwDN7V21DCQSXrkehzPZQr+zw=</saml2p:Artifa
ct>
</saml2p:ArtifactResolve>
```

Sample Artifact Resolve Response

The example below shows a sample artifact resolve response where HTTP Artifact is used as the binding for the SAML response.


```
<samlp:ArtifactResponse Version="2.0" ID="Lr3PBW2qy02RJhUtnBS2Su1ER7G" IssueInstant="2014-10-
20T09:54:18.499Z" InResponseTo="_44213af2e2143e460bbaab99c5f3d76c"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
  <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://inlvm10:9031</saml:Issuer>
  <samlp:Status>
    <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </samlp:Status>
  <samlp:Response Version="2.0" ID="T3fgEsFwWAr_b8HThq1ps4i8Kju" IssueInstant="2014-10-20T09:54:18.106Z"
InResponseTo="_822783897a4a2e30634b66803006b177">
    <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://inlvm10:9031</saml:Issuer>
    <samlp:Status>
      <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
    </samlp:Status>
    <saml:Assertion ID="ffOpZU94kDaPB9b5Iu7BrdHmpj6" IssueInstant="2014-10-20T09:54:18.110Z" Version="2.0"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
      <saml:Issuer>https://inlvm10:9031</saml:Issuer>
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
          <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
          <ds:Reference URI="#ffOpZU94kDaPB9b5Iu7BrdHmpj6">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>GM/ZGCR/g7Is6yWNo5DngRE8vRw=</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>

<ds:SignatureValue>TooIGEF0OF9ZiSdUS+1I2VUB5UfLbOURKfb2csFshh/+kE6tUD1ITB5CWwMVPYxcxGKGNP+e
gak2
xB0KP4RGd9KAhP7iMW+XGydyaIWkIwZJW9wX9fV4tscXHREp1cqB6pEiFrqfS0gCb88cEhNVdiUB
lSRb/wvblELZNPloH5k=</ds:SignatureValue>
      </ds:Signature>
      <saml:Subject>
        <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
NameQualifier="https://inlvm10:9031"
SPNameQualifier="sp.redirect.in">9518405DBA65D46B61D26C6302F885FD7018FB2C</saml:NameID>
        <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
          <saml:SubjectConfirmationData Recipient="http://example.com:9900/api/login/ssoLogin" NotOnOrAfter="2014-
10-20T09:59:18.110Z" InResponseTo="_822783897a4a2e30634b66803006b177"/>
        </saml:SubjectConfirmation>
      </saml:Subject>
      <saml:Conditions NotBefore="2014-10-20T09:49:18.110Z" NotOnOrAfter="2014-10-20T09:59:18.110Z">
        <saml:AudienceRestriction>
          <saml:Audience>sp.redirect.in</saml:Audience>
        </saml:AudienceRestriction>
      </saml:Conditions>
      <saml:AuthnStatement SessionIndex="ffOpZU94kDaPB9b5Iu7BrdHmpj6" AuthnInstant="2014-10-
20T09:54:18.110Z">
        <saml:AuthnContext>

<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified</saml:AuthnContextClassRef>
```

```
      </saml:AuthnContext>
    </saml:AuthnStatement>
    <saml:AttributeStatement>
      <saml:Attribute Name="Email" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
        <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">jane.saoirse@example.com</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute Name="FirstName" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
        <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">Jane</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute Name="LastName" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
        <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">Saoirse</saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
  </saml:Assertion>
 </samlp:Response>
</samlp:ArtifactResponse>
```


Sample Assertion

The example below shows a sample SAML Assertion.


```
<samlp:Response Version="2.0" ID="hrYt69818r5Hy0Ybr3SL6u.UF22" IssueInstant="2014-10-20T09:49:53.729Z"
InResponseTo="_feff8076a12bfacfbfd46528adc0f410" xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
 <saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">https://inlvm10:9031</saml:Issuer>
 <samlp:Status>
  <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
 </samlp:Status>
 <saml:Assertion ID="CD7IMSINXAUryvW2-WNPkcaFDFd" IssueInstant="2014-10-20T09:49:53.732Z"
Version="2.0" xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
  <saml:Issuer>https://inlvm10:9031</saml:Issuer>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="#CD7IMSINXAUryvW2-WNPkcaFDFd">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>F5S/9xcA7+zayq3ngJvCU9G5Wdg=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>

<ds:SignatureValue>lS8M4EfwH687yjmg2UeVYL7R/GMFv1akSaKpUa54F9I30yV3XoEhOD/prei5wiIxJCyjTszjtExd
NX8L7SpMKreqDBYu2gXQZfbydLxR/ugk5SySh4ZP/teAXvUU6/Qu8Mu8s047lo2eeNogiBlVDEc6
QAJZ9qiRq8/XpOPYrq4=</ds:SignatureValue>
  </ds:Signature>
  <saml:Subject>
    <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
NameQualifier="https://inlvm10:9031"
SPNameQualifier="sp.redirect.in">9518405DBA65D46B61D26C6302F885FD7018FB2C</saml:NameID>
    <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
```

```
      <saml:SubjectConfirmationData Recipient="http://example.com:9900/api/login/ssoLogin" NotOnOrAfter="2014-
10-20T09:54:53.733Z" InResponseTo="_feff8076a12bfacfbfd46528adc0f410"/>
    </saml:SubjectConfirmation>
  </saml:Subject>
  <saml:Conditions NotBefore="2014-10-20T09:44:53.733Z" NotOnOrAfter="2014-10-20T09:54:53.733Z">
    <saml:AudienceRestriction>
      <saml:Audience>sp.redirect.in</saml:Audience>
    </saml:AudienceRestriction>
  </saml:Conditions>
  <saml:AuthnStatement SessionIndex="CD7IMSINXAUryvW2-WNPkcaFDFd" AuthnInstant="2014-10-
20T09:49:53.732Z">
    <saml:AuthnContext>

<saml:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified</saml:AuthnContextClassRef>
    </saml:AuthnContext>
  </saml:AuthnStatement>
  <saml:AttributeStatement>
    <saml:Attribute Name="Email" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
      <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">jane.saoirse@example.com</saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute Name="FirstName" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
      <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">Jane</saml:AttributeValue>
    </saml:Attribute>
    <saml:Attribute Name="LastName" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified">
      <saml:AttributeValue xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">Saoirse</saml:AttributeValue>
    </saml:Attribute>
  </saml:AttributeStatement>
 </saml:Assertion>
</samlp:Response>
```

Chapter 5 | Identity Provider Configuration Examples

This chapter provides examples of configuring the Service Provider account for the following Identity Providers:

Identity Provider Configuration Example: SSO Circle

If you are using SSO Circle as your SAML Identity Provider, you can set up the domain in Policy Manager and then configure your Service Provider account in SSO Circle at www.ssocircle.com. Since SSO Circle's Identity Provider metadata file is publicly available, you can copy and paste the metadata when setting up the domain in Policy Manager.

To set up SSO Circle: high-level procedure

1. In Policy Manager, set up SSO Circle as an Identity Provider. See To set up SSO Circle as an Identity Provider in Policy Manager below.
2. In Community Manager, complete the setup by following the applicable procedure, depending on how you will be using the SAML domain:
   - As a login domain: see To enable a SAML login domain in Community Manager on page 18.
   - As an OAuth provider domain: see To configure a SAML OAuth Provider domain in Community Manager on page 18.
3. In your SSO Circle account, set up the platform as a Service Provider. See To set up the platform as a Service Provider using SSO Circle as the Identity Provider on page 40.
4. In Community Manager, test to make sure your domain that uses the SAML Web Browser SSO feature works correctly:
   - As a login domain: see Testing the SAML Domain as a Login Domain on page 19.
   - As an OAuth provider domain: see Testing the SAML Domain as an OAuth Provider Domain on page 19.

Prerequisites:

- Create an SSO Circle account at http://idp.ssocircle.com.
- In the SOA Software Admin Console, install the SAML feature. See Step 1: Install the Plug-Ins to Support SAML Web Browser SSO on page 13.
- In Policy Manager, set up your PKI keys.

To set up SSO Circle as an Identity Provider in Policy Manager

1. Log in to the Policy Manager Console.

2 Click the **Configure** tab, click **Security**, and then click **Identity Systems**.

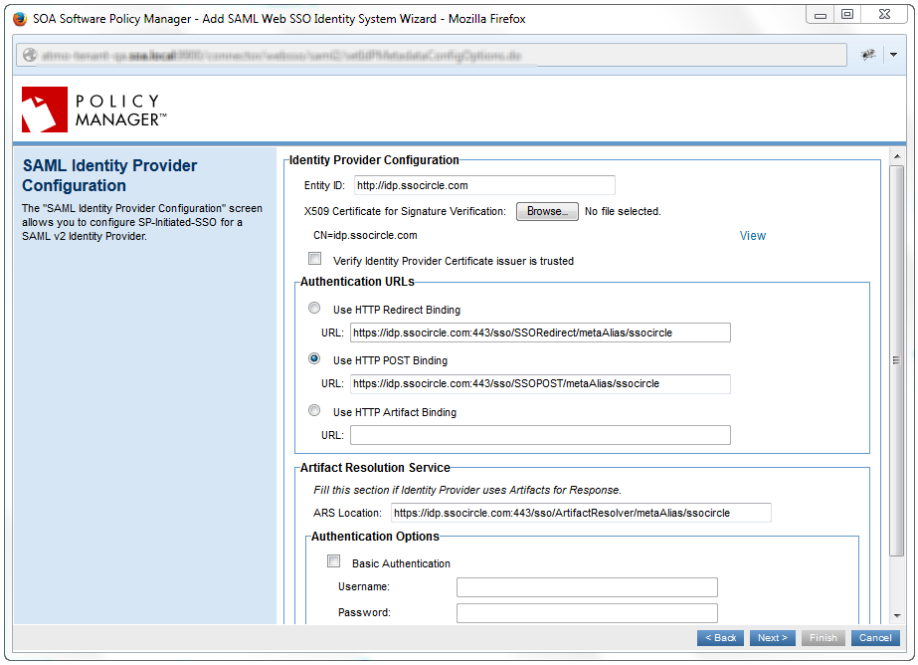
3 Click **Add Identity System** to access the Add Identity System wizard.

4 In the first page of the wizard, for identity system type, choose **SAML Web Browser SSO**. Provide name and description, and then click **Next**.

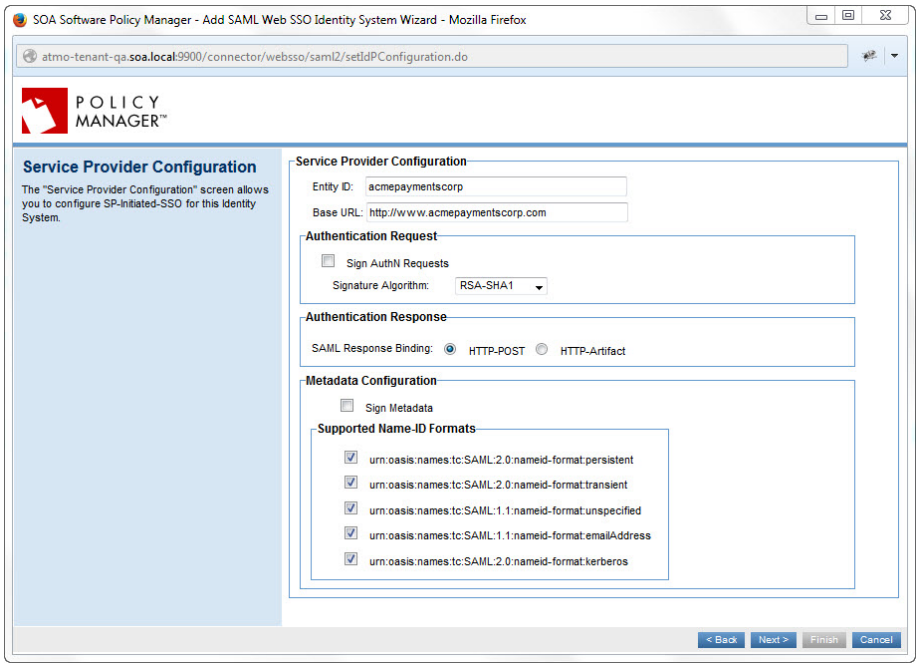
5 In the second page of the wizard, **Select SAML Identity Provider Configuration Method**, choose to configure using the metadata document, and enter the metadata URL for SSO Circle: http://idp.ssocircle.com.

6 Click **Next** to access the SAML Identity Provider Configuration page, populated with the SSO Circle values as shown below.

Note: If you chose to manually configure, you would need to enter the values on this page.



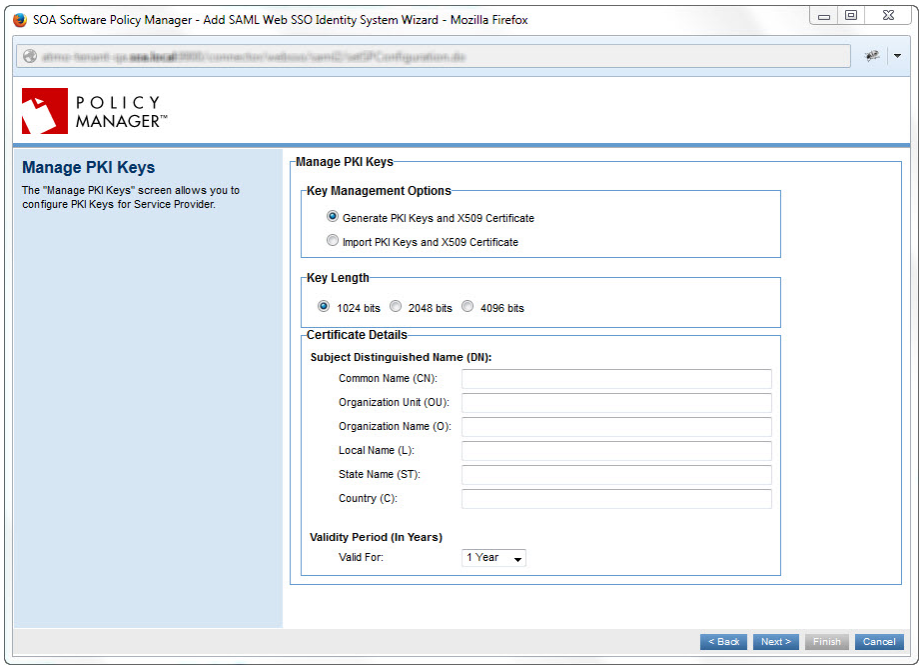
7 Review the values, change the binding type if needed, and then click **Next** to access the Service Provider Configuration page, as shown below.



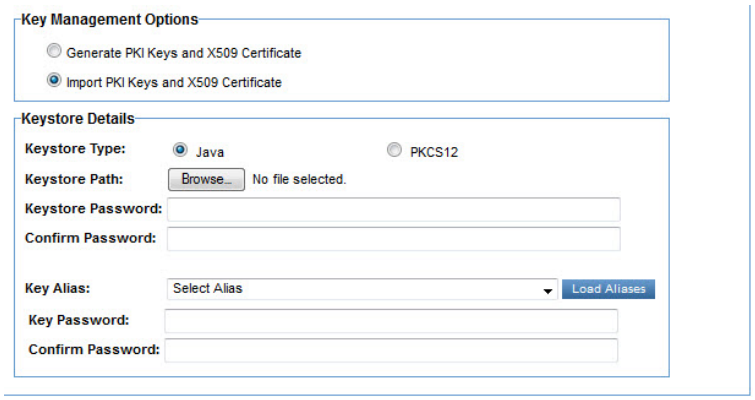
8 Enter the Service Provider configuration values, as needed:

- **Entity ID**: A unique ID that you define yourself, to identify your Service Provider in the SAML authentication request messages. When setting up your account with the Identity Provider you must specify the Entity ID, which must be unique within the IdP so that the IdP can identify your Service Provider; then, you set up the same value in Policy Manager.
- **Base URL**: used to construct the default Assertion Consumer Service (ACS) endpoint, the endpoint where the Service Provider will receive SAML assertions from the Identity Provider. Must be the container address of the container where the SAML Web SSO feature is initialized (<protocol_scheme>://<host>:<port>). For more information, see Base URL on page 56.
- **Authentication Request**: Generally, you would choose to sign authentication requests.
- **Authentication Response**: choose from the two supported bindings.
- **Sign Metadata**: SSO Circle does not support signing of the metadata, so leave this box cleared.
- **Supported Name-ID formats**: all are checked by default.

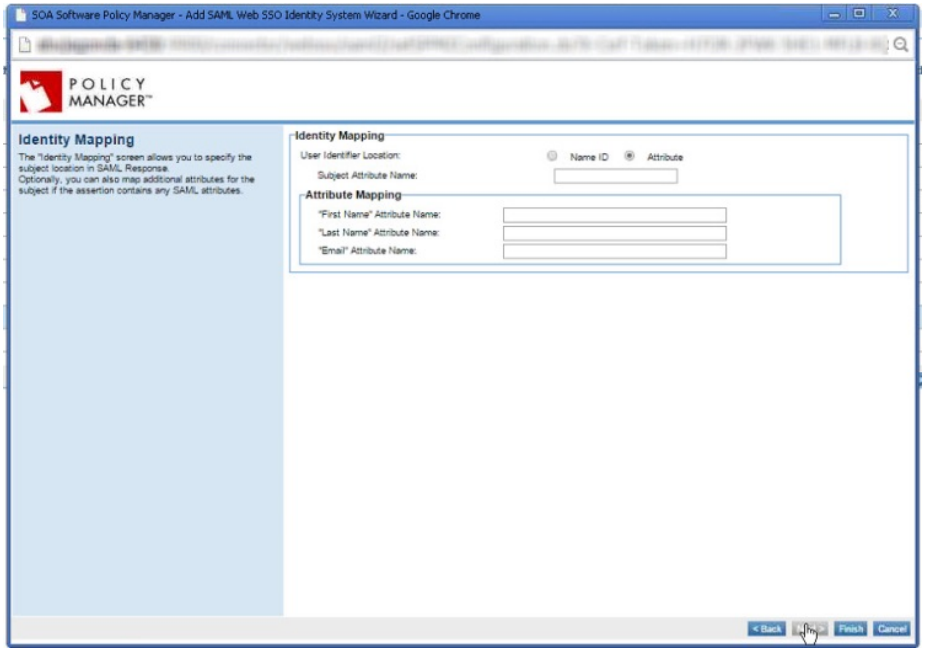
9 Click **Next** to access the **Manage PKI Keys** page as shown below. Here you will set up the keys you will use to sign your SAML authentication request messages.



10 Choose to generate or import keys. If you choose **Generate**, provide values in the **Certificate Details** section. If you choose **Import**, you'll need to choose a key management option and provide keystore details as shown below.

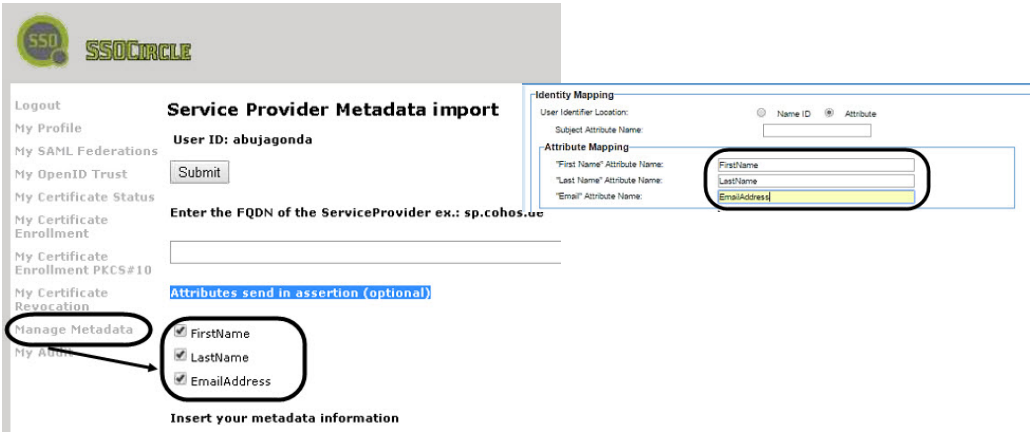


11 Specify additional key values if needed, and then click **Next** to access the Identity Mapping page as shown below.



12 Choose whether to send the NameID as the subject of the SAML assertion, or to use an attribute: if needed, define the subject attribute name.

13 Set up the Attribute Mapping values to correspond with those set up in your SSO Circle account, as shown below.



Note: SSO Circle supports only the above three attributes.

14 Click **Finish**. The Service Provider metadata file is generated and is available at the following URL: <http(s)>://<hostname>:<port>/saml/<sp_domain_name>/metadata. An example is shown below.

You will use this to set up your Service Provider account with SSO Circle, as shown in the next procedure.

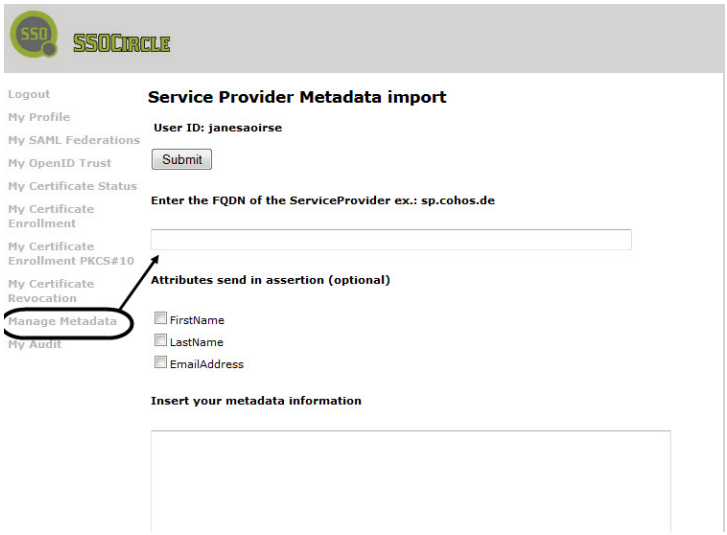
Note: Make sure you get the metadata.xml file for the container that has Community Manager installed (if the domain will be used for Community Manager login) and/or has the OAuth Provider feature installed (if the domain will be used for Community Manager OAuth domain, for resource owner authentication for at least one OAuth Provider).

To set up the platform as a Service Provider using SSO Circle as the Identity Provider

1 Log in to your account at www.ssocircle.com.

2 On the left, click **Manage Metadata**.

3 Click **Add New Service Provider**. The page looks something like the below.



4 Provide the following information:

- **FQDN of the Service Provider**: This value must match the Entity ID in the Policy Manager domain setup, **Service Provider Configuration** tab.
- **Attributes**: Choose one or more out of the supported attributes displayed. With SSO Circle, you cannot modify attributes.
- **Metadata information**: Copy the content of the metadata.xml file generated as a result of your domain setup (see the previous procedure). Be sure to **exclude** the XML processing instruction at the beginning of the file. Paste it in the box.
  Note: Do not copy the XML from the browser window. Instead, click View Source and copy it from there, excluding the XML processing instruction at the top.

5 Click **Submit**. The XML is processed and you should see a Success message when processing is complete. If there are any errors, review your information and make sure all the values were set up correctly in Policy Manager, then try again.

Note: If you make any changes, such as adding an OAuth Provider Domain in Community Manager, remember to update your SSO Circle account by pasting the revised metadata XML. If you are using the free SSO Circle account, you cannot edit your account; you'll need to delete the existing SP Metadata instance and then create a new one with the same name.

To configure and test in Community Manager

Once the setup in Policy Manager and the Identity Provider is complete, the steps to configure and test in Community Manager are the same for all Identity Providers. Follow the steps given earlier in this publication:

- Community Manager configuration: see Step 6: Community Manager Configuration on page 17.
- Testing: see Step 6: Test on page 19.

Identity Provider Configuration Example: PingFederate

This section provides additional information regarding setting up your Service Provider(SP) in PingFederate. In some cases, screen captures are provided; these are taken from PingFederate version 7.1.3.1 and are offered only as examples.

When registering with PingFederate as the SAML Identity Provider (IdP), note the following:

- To set up the domain in Policy Manager, you must provide the metadata of the IdP. However, with PingFederate, the IdP metadata is not available until you have set up your Service Provider details in PingFederate. This is because PingFederate allows customization of the IdP for a specific Service Provider account, so the IdP metadata file is created as a result of configuring your account, rather than released as a static file as the SSO Circle file is. Therefore, you must do one of the following:
  a) Create the SP connection in PingFederate first, export the metadata XML file, and then create the domain in Policy Manager.
  b) Configure the domain manually in Policy Manager, export the metadata XML file, then import the file into the PingFederate account.

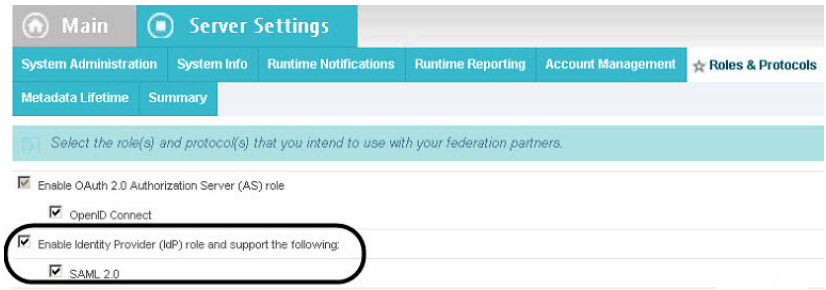
  When setting up the domain manually, you have to assume some default values, and then use the same values when you set up the Service Provider connection in PingFederate.

  The example given here shows manual configuration of Policy Manager as per b) above.

- If the security key you use for the account is issued by a third-party CA, you'll need to set up the CA as a trusted key issuer in Policy Manager.

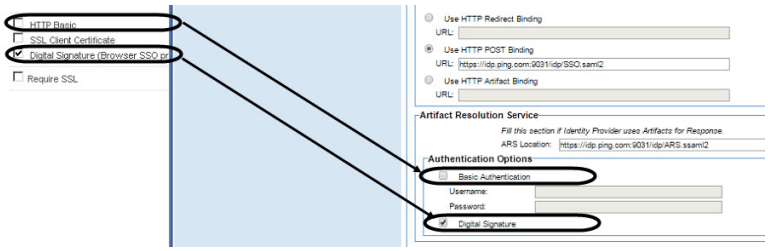
Prerequisites:

- In PingFederate, choose **Server Settings** > **Roles & Protocols**. Click the checkboxes to enable Identity Provider role and support SAML 2.0, as shown below.

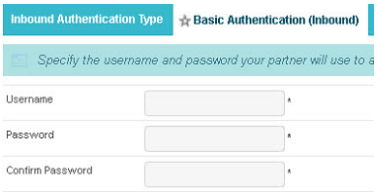


- In PingFederate, choose **Federation Settings** > **Protocol Endpoints**. Set up the applicable endpoint for Single Login service, and copy the binding URL (either Redirect or POST). You will use this in your Policy Manager setup.
- Conditional—if using HTTP Artifact for response binding: In PingFederate, when configuring security settings for the Artifact Resolution Service (Service Provider Credentials > Credentials > Configure, only available if HTTP Artifact is enabled), make sure you choose either HTTP Basic or Digital Signature for the authentication methods, and make sure that your choice matches you choice in Policy Manager setup, **SAML Identity Provider Configuration** page, as shown below.

The platform does not support the SSL Client Certificate option.



In PingFederate, if the artifact resolution service is used, if you choose HTTP Basic authentication, you'll need to set up the username and password in the Basic Authentication (Inbound) tab that appears when you choose that option, as shown below.



To set up PingFederate: high-level procedure

1 In Policy Manager, set up PingFederate as an Identity Provider. See To set up PingFederate as an Identity Provider in Policy Manager (Manual Configuration) below.

2 In your PingFederate account, set up the platform as a Service Provider. See To set up the platform as a Service Provider using PingFederate as the Identity Provider on page 47.

3 In Community Manager, complete the setup by following the applicable procedure, depending on how you will be using the SAML domain:

- As a login domain: see To enable a SAML login domain in Community Manager on page 18.
- As an OAuth provider domain: see To configure a SAML OAuth Provider domain in Community Manager on page 18.

4 In Community Manager, test to make sure your domain that uses the SAML Web Browser SSO feature works correctly:

- As a login domain: see Testing the SAML Domain as a Login Domain on page 19.
- As an OAuth provider domain: see Testing the SAML Domain as an OAuth Provider Domain on page 19.

To set up PingFederate as an Identity Provider in Policy Manager (Manual Configuration)

1 Log in to the Policy Manager Console.

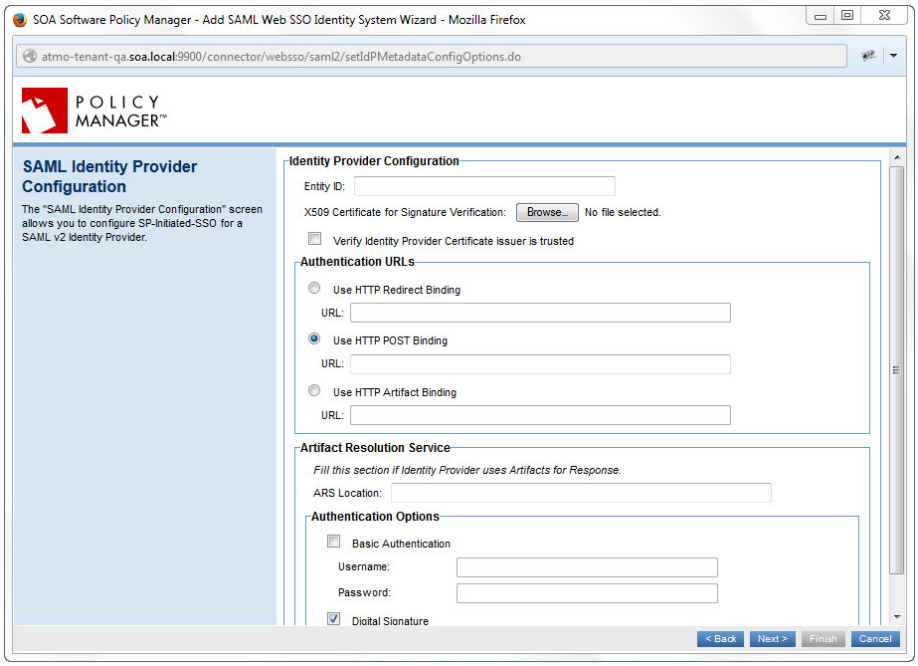
2 Click the **Configure** tab, click **Security**, and then click **Identity Systems**.

3 Click **Add Identity System** to access the Add Identity System wizard.

4 In the first page of the wizard, for identity system type, choose **SAML Web Browser SSO**. Provide name and description, and then click **Next**.

5 In the second page of the wizard, **Select SAML Identity Provider Configuration Method**, choose to configure manually.

6 Click **Next** to access the **SAML Identity Provider Configuration** page, as shown below.

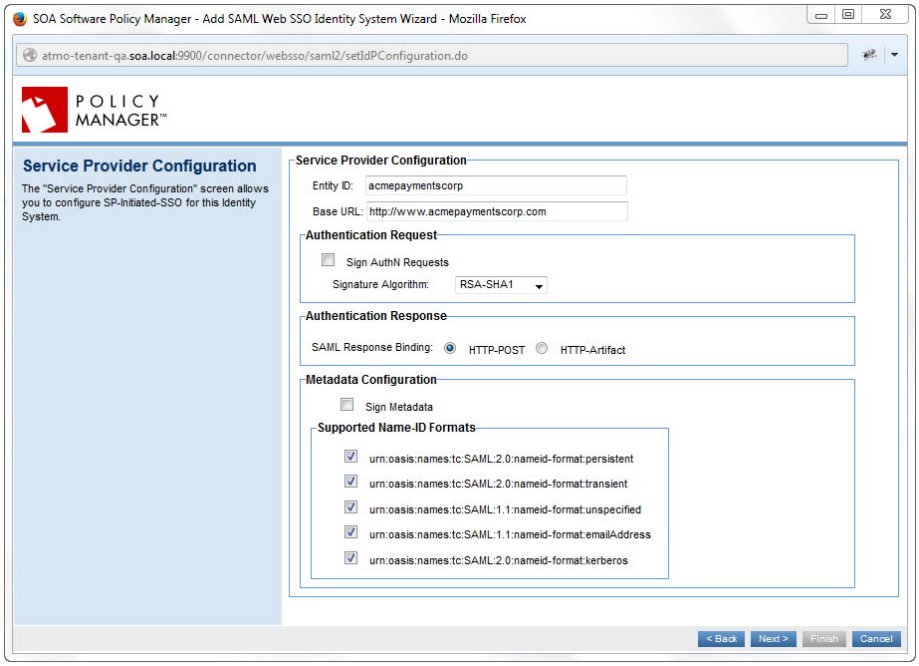


7 Enter the following values:

- IdP **Entity ID**: Make sure the unique Entity ID for the Identity Provider matches with the value in PingFederate > Server Settings > Federation Info > SAML 2.0 Entity ID. For example, idp.ping.com.
- **X.509 Certificate for Signature Verification**: The certificate that the IdP will use to sign the SAML response. Make sure it matches with the value in PingFederate > Security > Digital Signing & XML Decryption Keys & Certificates. You can use an existing certificate in PingFederate or create a new one, export it, and then upload it to Policy Manager.
- **Verify Identity Provider Certificate issuer is trusted**: If the issuer is PingFederate, the issuer and subject are the same, so this is not needed; but if the certificate is issued by a CA, it's best to check this box. You would also need to set up the CA as a trusted key issuer in Policy Manager (Configure > Security > Certificates > Trusted CA Certificates).
- **Authentication URLs**: Choose the binding you will be using, either HTTP POST or HTTP Artifact, and put in the applicable URL from your PingFederate account (Federation Settings > Protocol Endpoints, SSO Service section). Make sure both values match exactly. For example, let's say the value in your PingFederate setup is /idp/SSO.saml2, and the PingFederate Base URL is https://idp.ping.com:9031 (https://<host>:<port>); the URL might be https://idp.ping.com:9031/idp/SSO.saml2.
  Note: The platform currently does not support HTTP Artifact Binding for SAML Web Browser SSO requests.
- **Artifact Resolution Service**: Although the platform does not support HTTP Artifact for authentication request messages, it does support this binding for response messages. If you want to use it, you must configure the Artifact Resolution Service (ARS). Make sure the values match your PingFederate setup (Federation Settings > Protocol Endpoints, Artifact Resolution Service field).

– **ARS Authentication**: If you are using HTTP Artifact you will need to specify authentication for the Artifact Resolution Service: Basic Authentication, Digital Signature (of the Service Provider), or both.

8 Click **Next** to access the Service Provider Configuration page as shown below.



9 Enter the Service Provider configuration values. Note:

– **Entity ID**: A unique ID that you define yourself, to identify your Service Provider in the SAML authentication request messages. When setting up your account with the Identity Provider you must specify the Entity ID, which must be unique within the IdP so that the IdP can identify your Service Provider.

– **Base URL**: Used to construct the default Assertion Consumer Service (ACS) endpoint. Must be the container address of the container where the SAML Web Browser SSO feature is initialized (<protocol_scheme>://<host>:<port>). For more information, see Base URL on page 56.

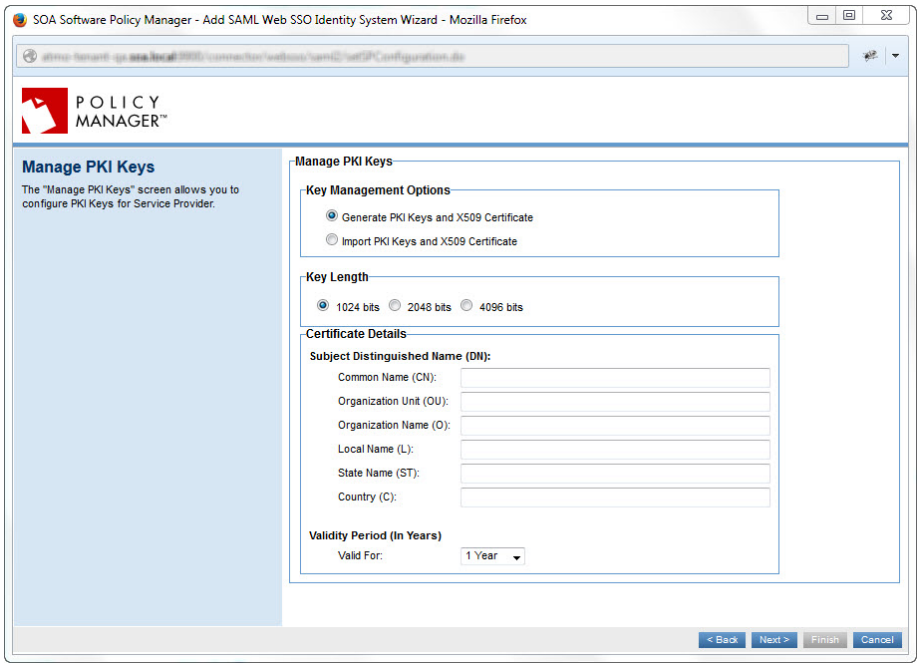
– **Sign Authn Requests**: Generally, it's best to sign authentication requests.

– **Authentication Response**: choose from the two supported bindings.
Note: If you are using HTTP Artifact, you must configure the Artifact Resolution service (previous page of wizard).

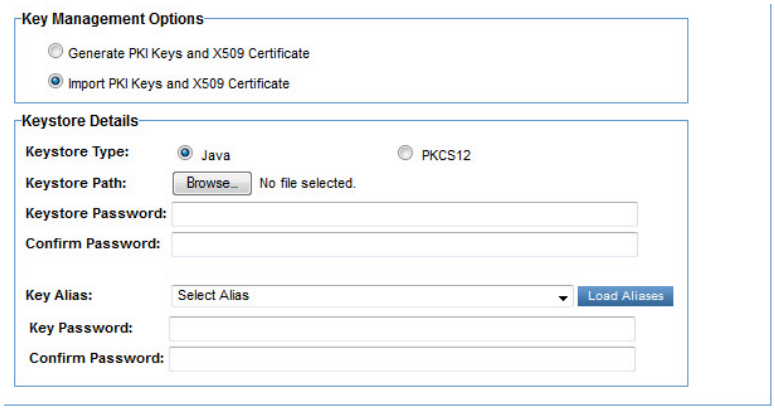
– **Metadata Configuration**: Choose whether or not to sign the metadata.

– **Name-ID Formats**: select all.

10 Click **Next** to access the **Manage PKI Keys** page as shown below. Here you will set up the keys you will use to sign your SAML authentication request messages.



11 Choose to generate or import keys. If you choose **Generate**, provide values in the **Certificate Details** section. If you choose **Import**, you'll need to choose a key management option and provide keystore details as shown below.

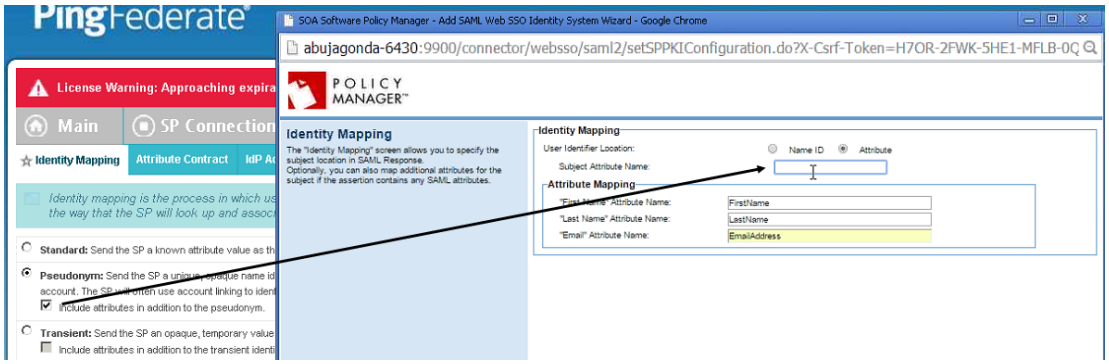


12 Specify additional key values if needed, and then click **Next** to access the Identity Mapping page.

13 Set up appropriate values for your PingFederate account:

- **User Identifier Location**: Specify whether the user identifier is part of NameID or Attribute. This corresponds to the Identity Mapping setup in PingFederate (first example below).
- **Attribute Mapping**: Make sure the values match those set up in your PingFederate account.

Examples are shown below.



14 Click **Finish**. The identity system is created in Policy Manager and the Service Provider metadata file is generated.

15 View the metadata.xml file in the browser at the following URL:

– <http(s)>://<hostname>:<port>/saml/<sp_domain_name>/metadata

Note: Make sure you get the metadata.xml file for the container that has Community Manager installed (if the domain will be used for Community Manager login) and/or has the OAuth Provider feature installed (if the domain will be used for Community Manager OAuth domain, for resource owner authentication for at least one OAuth Provider).

16 Save the metadata.xml file to your local drive so that you can upload it to PingFederate.

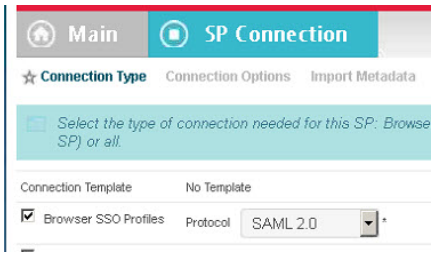
To set up the platform as a Service Provider using PingFederate as the Identity Provider

Once you've created the identity system, as explained in *To set up PingFederate as an Identity Provider in Policy Manager (Manual Configuration)* above, the platform creates the Service Provider metadata file. With this, you can set up your Service Provider account with PingFederate.

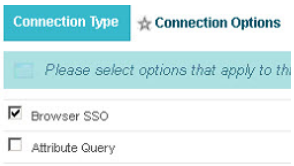
1 Log in to your account in the PingFederate Admin Console.

2 In PingFederate, under SP Connections, click **Create New**.

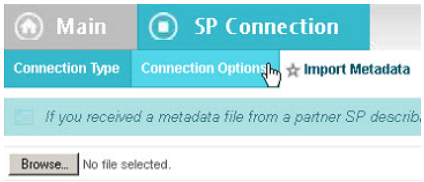
3 For Connection Type, choose Browser SSO Profiles, and then choose SAML 2.0, as shown below. Click **Next**.



4 At the **Connection Options** page, make sure Browser SSO is selected, and then click **Next**.



5 At the **Import Metadata** page, browse to the location of the Service Provider (SP) metadata.xml file generated by Policy Manager, which you exported at the end of the previous procedure, and import it. Click **Next**.



6 At the **General Info page**, review the settings taken from the SP metadata file you exported from Policy Manager and Imported to PingFederate. Click **Next**.

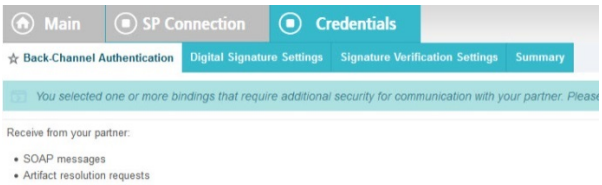
7 At the **Browser SSO** tab, click Configure Browser SSO, and then choose SP-Initiated SSO as shown below.

Set additional values:

- Assertion lifetime: accept or modify the values
- Click **Configure Assertion Creation** to configure at least one adapter mapping.

8 At the **Credentials** tab, shown below, there are three sets of credentials to configure:

- **Back-Channel Authentication**: If applicable, configure the inbound authentication option for the artifact resolution service at the IdP.
- **Digital Signature Settings**: configure the signature certificate for the IdP, for the response (SAML assertion). Include PKI keys, not just the certificate.
- **Signature Verification Settings**: configure the certificate for signature verification, for the IdP to verify the SP's signature on authentication request messages.



9 At the **Activation & Summary** page, review the summary and click through to the very end of the wizard to save all the values.

To configure and test in Community Manager

Once the setup in Policy Manager and the Identity Provider is complete, the steps to configure and test in Community Manager are the same for all Identity Providers. Follow the steps given earlier in this publication:

- Community Manager configuration: see Step 6: Community Manager Configuration on page 17.
- Testing: see Step 6: Test on page 19.

Chapter 6 | Modifying an Existing SAML Installation

Once you have everything set up correctly, your installation will be able to send SAML authorization request messages to your Identity Provider and receive SAML assertions or Artifacts in response.

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

These changes are essentially anything that affects a URL used for the SAML feature; either adding or changing the URL. For example:

- Creating a new OAuth Provider domain in Community Manager
- Changing the Authorization Server URL currently used for an OAuth Provider domain in Community Manager
- Configuring a new login URL for Community Manager

This chapter includes information about the steps you'll need to take to update your Service Provider account information with your Identity Provider if any of these changes occur.

Adding a New OAuth Provider Domain: Manual IdP Configuration

When you set up the domain in Policy Manager and generate the metadata XML file, there are two endpoints in the metadata XML file, in the AssertionConsumerService node. The first entry is designated as the default binding. An example is shown below.

```
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="http://acmepaymentscorp.com/saml/ACS/default" index="0" isDefault="true" />
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="http://acmepaymentscorp.com/api/login/ssoLogin" index="1" />
```

You then import the metadata to set up your Service Provider account with your Identity Provider.

However, let's say that in Community Manager you go on to set up one or more OAuth provider domains referencing this domain, as explained in this document. Each time a new OAuth provider domain is set up, the metadata XML file is updated dynamically with the endpoint for the new domain; but it will not work with the Identity Provider unless the new endpoint is set up on the identity provider side.

Each time a new OAuth Provider domain is set up in Community Manager, the new endpoint must be added in the Service Provider account with the Identity Provider so that the authorization endpoint for the new OAuth Provider domain will work.

Each Identity Provider has a different setup. Follow the applicable example below.

To add a new authorization URL in PingFederate

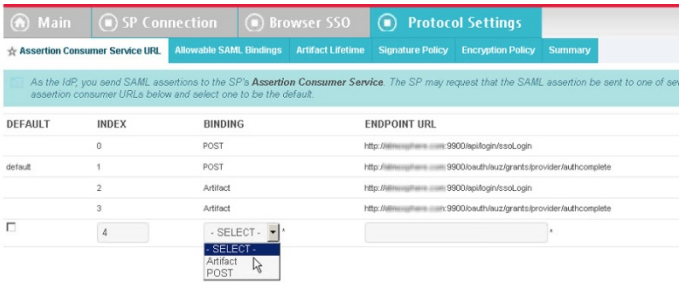
1 In Community Manager, create the new OAuth Provider domain. See To configure a SAML OAuth Provider domain in Community Manager on page 18.

2 Save the authorization server URL (set up on the last page of the wizard, Branding).

3 Sign in to your PingFederate account.

4 Go to Assertion Consumer Service URL.

5 Add the new URL, as shown below. Two values are important: Binding and Endpoint URL.



6 Save the changes.

To add a new authorization URL in SSO Circle

1 In Community Manager, create the new OAuth Provider domain. See To configure a SAML OAuth Provider domain in Community Manager on page 18.

2 Go to the updated metadata file (<http(s)>://<hostname>:<port>/saml/<sp_domain_name>/metadata). Make sure the new URL is there; refresh the page if needed.

3 Click View Source, and copy the XML, making sure you exclude the XML processing instruction at the top.

4 Sign in to your account in SSO Circle and click Manage Metadata.

5 Update the metadata for the account. If you are using the free SSO Circle service, you cannot edit the account; you will need to delete and recreate. Make sure you use exactly the same name.

6 Paste the updated metadata and click **Submit**. The XML is processed and you should see a Success message when processing is complete.

Chapter 7 | Troubleshooting

This section included information that might help you troubleshoot issues with your SAML single sign-on setup. It includes:

URL-unsafe characters in URL

The Community Manager domain name is used in the Service Provider metadata URL. The domain name must have URL-safe characters or must use escape characters.

If the domain name includes characters that are not URL-safe, it might not work correctly.

I can log in to Identity Provider but not to the developer portal

If you can the Identity Provider login screen, and login is successful, but you have not been successfully logged in to the developer portal, it could be due to one of a number of issues.

In the Policy Manager SAML Web Browser SSO domain, check the following:

- In the Identity Provider configuration, make sure the configured X.509 certificate for signature verification is still being used by the IdP to sign the assertions.
- If the IdP is encrypting the SAML response, make sure the IdP is using the certificate that is configured for Service Provider on the **Manage PKI Keys** page.
- In the **Identity Mapping** section, check to see that the user identifier location is configured correctly.
- For HTTP-Artifact binding, check the following:
  - If the IdP uses the HTTP-Artifact binding of the Service Provider to send the artifact reference, make sure the correct authentication options have been selected on the **SAML Identity Provider Configuration** page.

  - if the IdP Artifact Resolution Service is on HTTPS, make sure there are no SSL handshake errors. If there are errors, import the CA certificate of the IdP server certificate into the Policy Manager trust store.
- Check if the IdP uses valid Name-ID format to communicate the subject inside the SAML response. If needed, check allowed Name-ID formats in the **Service Provider Configuration** section.

I set up everything but I don't see the IdP login screen

In the Policy Manager SAML Web Browser SSO domain, do the following:

- Check if the Authentication URL configured on the **SAML Identity Provider Configuration** page is valid.
- Make sure the certificate that is configured on the **Manage PKI Keys** page is still used at the IdP for verifying the signatures on incoming AuthNRequests from this Service Provider.
- If this issue is with end-user login in OAuth Provider, make sure the SSO login URL of this OAuth Provider is registered at the IdP as a valid Assertion Consumer Service URL with correct binding format.
- Check if the SAML Service Provider domain is initialized without any errors.

How can I change the IdP login screen from main page to popup or vice versa?

When enabling a domain for login in Community Manager, you can choose whether the login page is displayed as the main page or as a popup.

Log in as the Site Admin, go to Config > Logins, and then, in the Mode column, choose Popup or Main.

For more information, refer to the Community Manager online help: What login page integration modes are supported?

I can complete an end-to-end case but my developer portal screens looks different from the IdP login screen

The Community Manager user interface look and feel is controlled by the styles defined in the custom.less file, which is uploaded by the Site Admin. Default styles are provided with installation, and the Site Admin can customize as needed.

If the Identity Provider login screen is customizable, you will find this defined somewhere in the IdP account settings. For example, in PingFederate it's managed with an HTML adapter.

Some customers like to have a similar look and feel for all UI elements a user might see in the course of their platform experience; others prefer that the external login screen has a different look and feel, which conveys to the user that the login screen is external to the platform.

If you want a similar look and feel for both, you'll need to customize the Community Manager user interface, the Identity Provider login screen, or both.

I am doing the SP setup in PingFederate, but when I import the metadata.xml file I get the message "Invalid signature on metadata file." What do I do?

This message means that the signature is being validated and has been found to be invalid. If this occurs, check the following:

- In Policy Manager, in Service Provider configuration, make sure that the **Sign Metadata** checkbox is checked.
- In the next tab, **Manage PKI Keys**, check the certificate details that are displayed.
- In Policy Manager, make sure the CA is set up (Configure > Security > Certificates > Certificate Authority) and export the certificate.
- Then, in PingFederate, go to Trusted CAs and import the certificate.

I configured the OAuth provider and configured SAML Web Browser SSO domain for the resource owner domain, but could not log in

When you set up the domain in Policy Manager, generate the metadata XML file, and import the metadata file to your Identity Provider, the file includes the information about the endpoints you've configured.

When you add a new OAuth provider domain in Community Manager, a new endpoint is added to the metadata XML file. However, in order for this to work, you must add the information about the new URL to the Service Provider account with your Identity Provider. If this step is not done, the URL will not work.

For more information, and instructions, refer to Adding a New OAuth Provider Domain: Manual IdP Configuration on page 50.

The metadata file doesn't have the login URL for my Community Manager installation

When you create the identity system in Policy Manager, you must make sure you access the metadata.xml file by using the metadata URL of the container where the features are installed that are using the SAML Web Browser SSO feature. For example:

- If this domain is supporting Community Manager login, use the container that has the Community Manager feature installed.
- If OAuth Provider is using this domain for end user authentication, use the container that has the OAuth Provider feature installed.

If there is no container with all the features that use the domain installed, get the metadata XML from one of the containers and add the other SSO URLs manually to it before registering the Service Provider with Identity Provider.

If you provide the wrong metadata.xml file to the Identity Provider, the feature will not work.

For example, for login, you should see the login URL for your instance of the platform listed in one of the AssertionConsumerService nodes in the metadata.xml file, as shown below (variables for base URL shown in angle brackets).

```
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="<http(s)>://<hostname>:<port>//api/login/ssoLogin" index="7"/>
```

For OAuth Provider Domain, the entry in the AssertionConsumerService node would look something like the below.

```
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="<http(s)>://<hostname>:<port>/oauth/auz/grants/provider/authcomplete" index="8"/>
```

Appendix A | Glossary of Terms

assertion

See *SAML assertion.*

Assertion Consumer Service (ACS) endpoint

The endpoint where the Service Provider will receive SAML assertions issued by the Identity Provider (<Response> message if HTTP POST is used artifact if HTTP Artifact is used).

Artifact

See *SAML Artifact*.

Artifact Resolution Service (ARS)

A service that you must set up if you want to use the HTTP Artifact binding (supported only for single sign-on SAML response messages). In a response message scenario, the ARS is on the Identity Provider side. You can then use the service to retrieve the full message using the artifact. See HTTP Artifact below.

Attributes

One or more values you will use to identify your users with the Identity Provider. For example, you might use attributes of firstname, lastname, and email address, or you might use attributes of username and password.

Base URL

The base URL for your implementation: <protocol_scheme>://<host>:<port>. It must be the container address of the container where the SAML Web SSO feature is initialized and where the OAuth Provider feature is running (container or cluster URL). The platform uses this to construct the default endpoint, used for error responses.

Some Identity Providers, if an error is encountered, send the error message to the default URL specified in the Service Provider metadata file, rather than to the specific URL at which the error was encountered. For example, if there are two URLs in the file, one for platform login and the second for an OAuth Provider domain, an error message relating to the OAuth Provider domain would be sent to the platform login endpoint. PingFederate is an example of an Identity Provider that returns an error response in this way.

To get around this, the platform constructs a default endpoint to be use for error responses, using the base URL for your implementation.

For example, if the base URL is http://www.acmepaymentscorp, the platform would construct the following default endpoint: http://www.acmepaymentscorp/saml/ACS/default. This would show as the first md:AssertionConsumerService entry in the exported Service Provider metadata file generated when you configure the identity system entry in Policy Manager.

To view an example, see the sample Service Provider metadata file: Sample Metadata File: Service Provider on page 30.

Entity ID

A unique identifier for a SAML entity. A SAML entity can be a Service Provider or an Identity Provider.

As a Service Provider, you define your Entity ID. When setting up your account with the Identity Provider you must specify the Entity ID, which must be unique within the IdP so that the IdP can identify your Service Provider.

The Entity ID is used as the value of the <Issuer> element inside the SAML protocol message. In an authentication request, the <Issuer> element contains the Entity ID of the Service Provider; in the SAML response, it contains the Entity ID of the Identity Provider.

From the perspective of the Service Provider, the Entity ID is analogous to the client_id in OAuth.

HTTP Artifact

One of the binding options supported by the SAML protocol. HTTP Artifact is useful in scenarios where the SAML requester and responder are using an HTTP User-Agent and do not want to transmit the entire message, either for technical or security reasons. Instead, a SAML Artifact is sent, which is a unique ID for the full information. The IdP can then use the Artifact to retrieve the full information. The artifact issuer must maintain state while the artifact is pending. An Artifact Resolution Service (ARS) must be set up.

HTTP Artifact sends the artifact as a query parameter.

Community Manager currently supports this binding option for SAML responses, but not for SAML requests.

HTTP POST

One of the binding options supported by the SAML protocol.

HTTP POST sends the message content as a POST parameter, in the payload.

Community Manager currently supports this binding option for SAML, for both requests and responses.

HTTP Redirect

One of the binding options supported by the SAML protocol.

When HTTP Redirect is used, the Service Provider redirects the user to the Identity Provider where the login happens, and the Identity Provider redirects the user back to the Service Provider. HTTP Redirect requires intervention by the User-Agent (the browser).

HTTP Redirect sends the message content in the URL. For this reason it cannot be used for the SAML response, because the size of the response will typically exceed the URL length allowed by most browsers.

Community Manager currently supports this binding option for SAML requests.

Identity Provider

In terms of SAML, the Identity Provider is the entity that verifies the identity of the user, in response to a request by the Service Provider.

The Identity Provider is responsible for maintaining and authenticating the user's identity. In terms of platform usage, the Identity Provider verifies by means of user credentials such as username and password.

IdP

Abbreviation for *Identity Provider* (see above).

PingFederate

A third-party company that provides SAML Identity Provider services, verifying the identity of users for Service Providers using the SAML Web SSO protocol.

The CM SAML solution is tested with the PingFederate product. For more information, see https://www.pingidentity.com.

SAML

Acronym for Security Assertion Markup Language. SAML is an identity federation standard that enables single sign-on. It is an XML-based standard for exchanging authentication and authorization data between a Service Provider (providing a service to the user) and an Identity Provider (providing user identity verification for the Service Provider).

SAML Artifact

When the HTTP Artifact binding is used, the Artifact is a unique ID used by the Service Provider and Identity Provider to reference a specific user session or transaction. The SP can use the Artifact to query the IdP for information about the user.

SAML assertion

A SAML assertion is an XML document returned by the Identity Provider to the Service Provider after authentication of the user. The assertion has a very specific structure, as defined by the SAML standard. A SAML assertion has a <Subject> element which contains information about the user. It might have conditions and attributes associated with the information being conveyed. It is digitally signed and asserts that the user has been authenticated.

Note: the above definition applies to an authentication assertion, which applies in the context of the platform's support of SAML. There are other types of SAML assertions.

SAML Web SSO

Single sign-on over the Web using the SAML Web Browser SSO Profile. For references to the SAML standard for this profile, see SAML Specifications on page 9.

Service Provider

In terms of SAML, the Service Provider (SP) offers a service to the user and allows the user to sign in by using SAML. When the user attempts to sign in, the SP sends a SAML authentication request to the Identity Provider (IdP). The IdP validates the request, authenticates the user, and creates a SAML assertion that represents the user's identity and, in some cases, sends additional information about the user in the form of associated attributes. The SAML assertion is digitally signed and encrypted and then sent back to the Service Provider that initiated the request.

Identity federation software at the SP receives the assertion, verified the authenticity, decrypts, and shares the information with the application, which then logs in the user.

SSO

Abbreviation for single sign-on, a feature allowing a user to sign in once for more than one system rather than signing in separately to each system.

If an app offers single sign-on, this means that the app, acting as a Service Provider (providing services to an end user) uses an Identity Provider, an entity that provides authentication and possibly authorization services, to verify the identity of an end user logging on to the app. The user signs in to the Service Provider, and the Service Provider either implicitly or explicitly requests authentication from the Identity Provider. Once authentication is received, the Service Provider delivers the requested service to the end user.

SSO Circle

A third-party company that provides SAML Identity Provider services, verifying the identity of users for Service Providers using the SAML Web SSO protocol.

The CM SAML solution is tested with SSOCircle's SAML Identity Provider. For more information, see http://www.ssocircle.com.

SP

Abbreviation for *Service Provider* (see above).

–♦–