Single Sign-On with Apperian

Single sign-on (SSO) is an authentication method that allows users to log into multiple services using the same credentials. When you implement SSO with Apperian, your users can securely authenticate in the Admin Portal and App Catalogs using the same credentials they use throughout your enterprise.

> **How to Enable SSO**
>
> If you are interested in using SSO with Apperian, contact Customer Support. Most of the configuration is done by Customer Support, but you will need to provide some details (described below) about your SAML or OAuth implementation.

Using SSO has many benefits, including:

- strengthened corporate security
- tighter enterprise integration
- easier user provisioning
- improved user experience

With SSO authentication, Apperian never views or stores your users' login credentials.

> **SSO in the Admin Portal**
>
> SSO authentication in the Admin Portal is only available when users access your organization's vanity URL. For more information, see Vanity Subdomain.

When SSO is enabled, it becomes the authentication method used for both the Admin Portal and your App Catalogs; it can't be turned on for one and off for the other. However, SSO authentication in the Admin Portal is only available when users access your organization's vanity URL. So, you can effectively ignore SSO for the Admin Portal if you choose not to communicate the vanity URL to your users.

Apperian supports the following SSO protocols:

- SAML
- OAuth

Automatic Creation of Users and Groups

When Apperian authenticates a user through SSO (SAML or OAuth), a corresponding user is created in the Admin Portal if one does not already exist. This is known as auto-provisioning. During this process, new users are added to the "All Users" user group.

Apperian can also configure your organization to manage a user's group membership during SSO authentication as long as a Groups attribute is included with the user metadata. Apperian can add and remove a user from groups that are already defined in the Admin Portal (Group Matching). Optionally, Apperian can also create new groups and add the user to them (group auto-provisioning). For more information, see Group Assignment During SSO.

SSO through SAML

Security Assertion Markup Language (SAML) is an XML-based open standard for exchanging authentication data across the Internet. SAML uses security tokens containing "assertions" to pass information about a user between an authority (Identity Provider) and a consumer (Service Provider). Apperian supports SAML 2.0. For more information on the SAML standard, see SAML 2.0.

Here are some basic concepts that you should understand. SAML and OAuth use similar terminology, so the corresponding OAuth term appears in parentheses.

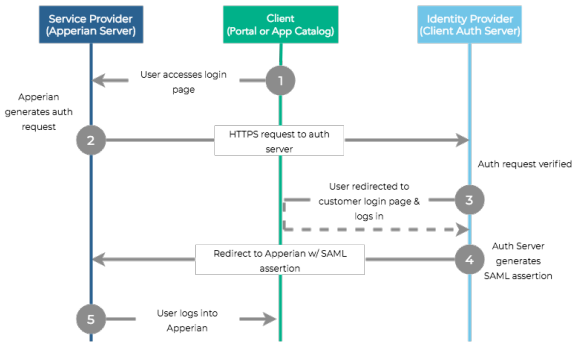
- **Service Provider (*Resource Server*)** - The web server that hosts a resource and provides access based on authentication information supplied by an Identity Provider. This is the Apperian cloud server.
- **Client** - The application used by an end user to interact with the Service Provider. This is the Admin Portal or an App Catalog.
- **Identity Provider (*Authorization Server*)** - The server that owns and maintains a directory of user credentials and an authentication mechanism. The Identity Provider authenticates a user and provides the Service Provider with a SAML assertion that confirms that authentication. This is the authentication server used throughout a customer's enterprise.
- **Assertion** - A packet of security information that can be processed by an authentication server.
- **PingFederate** - Cloud identity management software provided by Ping Identity®. Apperian runs PingFederate in the Apperian cloud and uses it to communicate with a customer's authentication server using SAML.
- **HTTPS** - Hypertext Transfer Protocol Secure (HTTPS) is a combination of HTTP and the SSL/TLS protocol which provides an encrypted and secure channel of communication over the Internet. Apperian exchanges HTTPS messages with PingFederate.

> When using SSO, your user authentication page must use *HTTPS*, not HTTP.

How it Works

The following diagram illustrates the SAML SSO flow.

During this process, Apperian never communicates directly with the authentication server. Instead, Apperian exchanges HTTPS messages with PingFederate which communicates with the authentication server using SAML.



1. A user launches the Admin Portal or App Catalog.
2. Apperian (the Service Provider) then generates an authorization request to the customer's authentication server (the Identity Provider). Apperian sends this HTTPS request to PingFederate. PingFederate sends a SAML request to the customer's authentication server.
3. The authorization server responds with a SAML assertion that includes a URL for the customer's authentication page. This authentication page can use whatever authentication method the customer wants.
4. Apperian displays a web view of the customer's Authentication page, rather than the default Apperian login page. The customer's authentication server processes the user's input and sends a SAML assertion when the user is authenticated.
5. Apperian then logs the user into the Admin Portal or App Catalog.

Enabling SSO Through SAML

To enable authentication through SAML:

1. **Provide SAML Metadata to Apperian.** To exchange SAML metadata with Apperian, export a SAML metadata file from your authentication server and send it to Apperian. A SAML metadata file provides configuration data that tells an Identity Provider and Service Provider how to establish a connection and communicate with each other. Your metadata file must provide authentication attributes and user attributes. For more information on the SAML metadata requirements, as well as details on how Apperian handles group assignment, see SAML Metadata.
2. **Test the SAML Connection.** Apperian will initiate a SAML connection with your authentication server and verify the content of the SAML assertions. If you do not wish to provide Apperian with test credentials, you can test the SAML connection yourself.

After you complete the steps above, Apperian will configure your organization for SSO authentication through SAML.

The App Catalog automatically updates its settings and begins using SAML SSO. If your App Catalog is already deployed to your users, you do not need to update and redeploy it.

SSO through OAuth

OAuth is an open standard for authorization that does not require users to share passwords. OAuth provides standard mechanisms to allow API clients to request and use tokens. Apperian supports OAuth 2.0. For more information on the OAuth Standard, see OAuth.

If your company uses OAuth, Apperian can configure the Admin Portal and App Catalog to route user authentication through OAuth. Rather than providing Apperian with secret user identity attributes (such as a user's password), the OAuth flow instead provides apperian with an access token that authenticates a user.

Apperian currently supports the following OAuth providers:

- LinkedIn
- Ping Identity
- Azure AD
- Azure AD B2C (**does not support biometric authentication**)

If you need to use a different OAuth provider, contact your Apperian account representative.

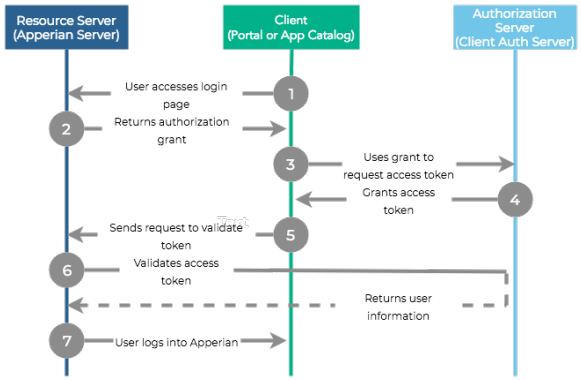
Here are some basic concepts that you should understand. OAuth and SAML use similar terminology, so the corresponding SAML term appears in parentheses.

- **Resource Server (*Service Provider*)** - The web server or API that requests and validates access tokens to provide authorization. This is the Apperian server. When the Resource Server responds to an access request, its API response must provide email and/or userid attributes. Apperian uses one or both of these attributes to match the identity with an existing user, or to provision a new user if one does not already exist. For more information on the user attributes in the API response, see OAuth Metadata.
- **Client** - The application used by an end user to interact with the Resource Server and request an access token. This is the Admin Portal or an App Catalog.
- **Authorization Server (*Identity Provider*)** - The server that is used to issue access tokens (and optionally refresh tokens, if supported by your OAuth provider). This is the customer's OAuth Provider. This server also hosts the OAuth provider's API.
- **User Agent** - A WebView (typically a browser-based authentication page) that Apperian opens and uses to redirect authentication requests.

⚠ Authorization through OAuth is not yet supported with the Windows 8/10 App Catalog.

How it Works

The following diagram illustrates the OAuth SSO flow.



1. A user launches the Admin Portal or App Catalog. Apperian (the client application) detects that the catalog is configured for OAuth and redirects to the Resource Server. If the user has never logged in before or has logged in but has an expired OAuth token, then the OAuth authorization screen appears.
2. Once the user successfully authenticates, the Resource Server returns an intermediate "authorization grant" code.
3. Apperian sends a token request, with the code, to the Authorization Server.
4. The Authorization Server exchanges the code for an OAuth access token, then sends this token to Apperian. If the OAuth provider supports refresh tokens, then the Resource Server can optionally provide a refresh token along with the access token. Depending on the OAuth provider, the Authorization Server may send the token in the Authorization Bearer Header of the response. Apperian can configure support for this in your organization.
5. Once Apperian receives this token, it sends an access request to the Resource Server.
6. The Resource Server contacts the Authorization Server to validate the token and returns information about the user, including `userid` and /or `email`, and possibly `firstname`, `lastname`, and `group` attributes.
7. Apperian then either matches that user to an existing user or auto-provisions a new user and then allows access to the Admin Portal or App Catalog.

Enabling SSO Through OAuth

To enable authentication through OAuth:

1. Add an application for the App Catalog on the Authorization Server. For the authorized redirect URL, enter the URL for your production environment.

Environment	Redirect URL
North America	https://easesvc.apperian.com/opentokenoauth.php
Europe	https://easesvc.apperian.eu/opentokenoauth.php

⚠ **Whitelist URLs**

Apperian recommends whitelisting these URLs as wildcards in your OAuth provider's configuration.

When you add an application, the OAuth provider will assign Authentication Keys (Client ID and Client Secret) to the application; you will need to provide these to Apperian in the next step.

ⓘ If you are working with a custom environment or not sure which environment to use, check with Customer Support.

2. Provide the following information to Apperian:

Authentication URL	The URL for the endpoint on the Authorization Server that will handle authentication requests, display the authentication page (the WebView), and return an authentication code after receiving valid login credentials.
Access Token URL	The URL for the endpoint on the Resource Server that will receive an authentication code and return an access token (and optional refresh token).
User Information URL	The URL for the endpoint on the Resource Server that provides information about the authenticated user (User, Email Address) so that Apperian can map this user to a user in the Apperian database. ⚠ If you're using Azure AD B2C, you do not need to provide a User Information URL.
Authorization Bearer Header	Inform Apperian if your Authorization Server sends the access token in the Authorization Bearer header of the response.
Client ID	These are the Authentication Keys assigned by the OAuth provider when you add an application for the App Catalog (step 1 above). Apperian will use these keys when it sends API requests to the Authentication Server.
Client Secret	
Scope	The permissions that Apperian will request on behalf of the user. You can specify multiple permissions.
Resource	**(For Azure AD only)** The URI of the protected resource that your application needs access to.
Instructions for Formatting API Requests /Responses	When Apperian communicates with the APIs hosted on your Authentication and Resource Servers, it needs to send requests in the proper format and know what responses from the APIs will look like. Provide Apperian with an example request and response for a token request to the Authorization Server and an access request to the Resource Server
OpenID Connect Configuration Endpoint	**(For Azure AD B2C Only)** Arxan uses this URL to retrieve the public key used to sign the JSON web token in order to verify that the tokens are valid. You can retrieve this URL in the **OpenID Connect configuration endpoint** from your Azure AD B2C implementation. It appears as follows: `https://{tenant}.b2clogin.com/{tenant}.onmicrosoft.com/{policy}/v2.0/.well-known/openid-configuration` For more information, see Validating tokens.

After you complete the steps above, Apperian will configure your organization for SSO authentication through your OAuth implementation.

✓ The App Catalog automatically updates its settings and begins using OAuth SSO. If your App Catalog is already deployed to your users, you do not need to update and redeploy it.