

API connector

Path to function: *Management > Connections > Add > API*

The OpenAthens local authentication API can be used to log your users into the system based on credentials stored in any system you can gain programmatic access to and is ideal in situations where you cannot use any of the other connection types. It requires you to implement some code at your end.

Your local systems should use at least two factor authentication (e.g. username and password, barcode and pin).

Preparation

Before you start you will need:

- An API Key you have generated for this - see [API keys](#)
- A member of your IT team with web-programming experience
- Access to the OpenAthens administration area at the domain level

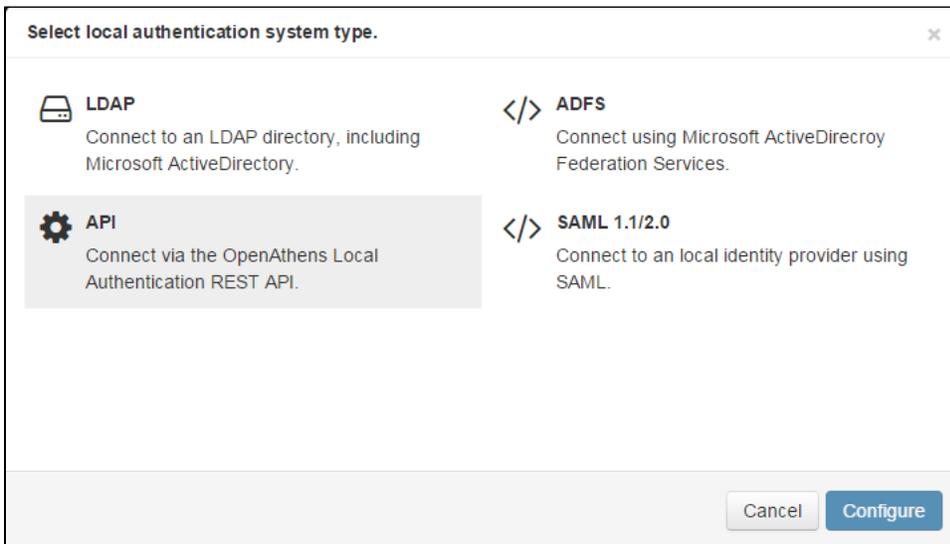
If you are migrating from an alternative IdP such as Shibboleth, also see: [Migrating from your own IdP](#)

If you're unsure about anything or get stuck, we're happy to help. Hit the support link in the top right of the admin area to get through to your local support guys.

Add the connection in OpenAthens

In the administration interface as the domain administrator go to *Management > Connections*

1. Click the add button on the left and select API



2. Enter a name for the connection. You can edit this later.
3. Enter a callback URL. This will be where users are sent to sign in if they do not already have an OpenAthens session when they attempt to access a resource - see [implementing the API in your code](#) for more detail. You can edit this address later.
4. Save changes

You should now see something similar to this:

The connection ID and URI are specific to your domain and generated automatically. You will need to pass both of these values, and the API key you generated to your IT colleague.

The Salt value is used only if you are [migrating from a SAML product such as Shibboleth or OpenAthens LA](#) and is unlikely to be necessary if you are adding this type of connector.

The detail fields displayed are

| Field | Explanation |
|-----------------------|---|
| Display name | The name of the connection as it will appear at our authentication point when there is a choice of connector. |
| Connection ID | The connection ID that needs to be specified in the API call payload. Pass this to your IT colleague before they begin their implementation. |
| Connection URI | The domain specific URI where the API will be expecting the POST. Pass this to your IT colleague before they begin their implementation. |
| Callback URL | An address in your site or application that users will be sent to for authentication (see: Implementing the API connector in your code) |
| Status | <p><i>Not live</i> = connection can only be used in debug mode. The visibility and default flags are ignored.</p> <p><i>Live and visible (if this is the only local connection)</i> = connection can only be used in debug mode.</p> <p><i>Live and visible (if there are multiple live and visible connections)</i> = users are offered a choice of connections, including this one. There is a domain preference to include OpenAthens accounts or not.</p> <p><i>Live & visible & default</i> = This is your only login option and users will be sent directly to your login whenever the organisation is known. A successful authentication will tell the authentication point to remember that location. A failed authentication will clear that setting. Debug mode will not show other login options.</p> <p>Changes to the status usually take effect within a few minutes.</p> |
| Create local accounts | <p>Automatically - any user authenticated by your system and passed back to us is deemed ok and will be accepted by the system</p> <p>Manually - only user IDs you have previously uploaded via the list page will be accepted by our systems</p> |
| Remove local accounts | <p>This setting controls when local account data will be automatically cleared from the system and is the number of days from the last time the account last signed in. Pre-mapped accounts that have not been seen are also cleared.</p> <p>The setting can be from 1 to 365 days and represents the number of complete days that have passed since the date the account last signed in. i.e. does not include the day of the last sign-in in the count. See also: How to modify a local account.</p> |

| | |
|------------|--|
| Salt value | <p>The salt used to generate a targetedID for users authenticated by this connection.</p> <p>You might edit it if you were migrating from something like Shibboleth so that your users can have the same targetedID value when they change systems. If you set it to blank the connection will use the same salt as your MD accounts.</p> <p>Modifying this after you go live will change the identifiers seen by service providers for your users... which is rarely desirable.</p> |
|------------|--|

Set up your application

See - [Implementing the API connector in your code](#)

Configure mappings and permission sets

The final two areas to configure are permission set rules and attribute mappings:

- [Permission set rules](#) so that your users are assigned an appropriate set of resources
- [Attribute mappings](#) so that OpenAthens can make use of data passed it by your source

Both of these functions can only look at data in the attributes array you pass.

When you're ready to go live, check both the live and visible boxes and then save. Your new connection should be available a few seconds later.

How to test

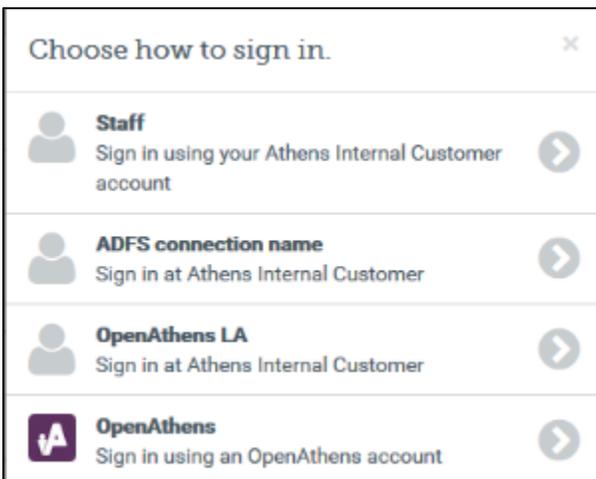
Discovery is not available until you set the connection as live and visible so that users do not get offered options that are not ready to be used. To test your connection you will need to use [debug mode](#) to make the connection selectable by you.

Once you have tested and are happy, you can set the connection as live, visible and optionally default then save. This will make it live for your users within a few seconds.

Multiple connectors and OpenAthens accounts

This type of connector is best used as the default connection. In this mode when a user arrives at our [authentication point](#) with your organisation known, such as would happen if they select it at a resource's login, use a WAYFless URL, [the Redirector](#) or have previously authenticated successfully, they are passed directly to your login without seeing our authentication point.

If you have a need to use multiple connections, or OpenAthens accounts alongside local accounts - e.g. if you have a group of users that are not in your directory - then you can set the connection as live and visible but not default and set it to allow OpenAthens accounts via the setting on the [domain preferences page](#). In this mode when the user arrives at the authentication point with your organisation known, they will initially see a chooser where they can select the connection to use - all live and visible local connections will be available as well as an option to use OpenAthens accounts. The authentication point will remember their choice.



Multi-valued attributes

With multi-valued attributes - e.g. the memberOf field in ADFS - the interface is not able to display all values and only display one. All values are read and cached though so are available for things like [permission set rules](#) and [attribute release](#).

Anything to watch out for?

Minimum TLS version is 1.2.

If you are planning to pre-upload user identifiers, you will need to have at least one local account visible in the list to access the upload button. Do not delete all your test logins until at least some of your pre-mappings are uploaded.

