OpenID Connect Authentication

- Video Guides
- Requirements & Limitations
- BookStack Configuration
- Authentication System Configuration
- Switching to OIDC with Existing Users
- Debugging
- Using a Different ID Claim
- Group Sync
  - Nested Groups Claim Example

OpenID Connect (OIDC) can be used within BookStack as a primary method of authentication. This replaces the default email & password authentication mechanism. BookStack supports a simple level of auto-discovery to ease endpoint and key management.

When used, BookStack will attempt to match the OIDC user to an existing BookStack user based on the "External Authentication ID" value stored against the Bookstack user. If this match cannot be made, BookStack will effectively auto-register that user to provide a seamless access experience. They will be given the default role set under the "Default user role after registration" option in the application settings.

Video Guides

We have a range of videos available that can help show, for a range of identity platforms, the exact steps required to set-up a BookStack OpenID Connect authentication integration:

- General setup guide (Using Okta as an example)
- Azure Active Directory OIDC Guide
- Authentik BookStack OIDC Guide

Requirements & Limitations

Listed below are some considerations to keep in mind in regard to BookStack's OIDC implementation:

- Only RS256 is currently supported as a token signing algorithm, Token encryption is not supported. This applies to both ID tokens and userinfo responses.
- Discovery covers fetching the auth & token endpoints, in addition to parsing any keys at the JWKS URI, from the `<issuer>/.well-known/openid-configuration` endpoint.
  - Issuer discovery is not supported.
- RP-initiated logout is supported if enabled, but any other logout mechanisms are not supported.

BookStack Configuration

To set up OIDC based authentication add or modify the following variables in your `.env` file:

```
# Set OIDC to be the authentication method
AUTH_METHOD=oidc

# Control if BookStack automatically initiates login via your OIDC system
# if it's the only authentication method. Prevents the need for the
# user to click the "Login with x" button on the login page.
# Setting this to true enables auto-initiation.
AUTH_AUTO_INITIATE=false

# Set the display name to be shown on the login button.
# (Login with <name>)
OIDC_NAME=SSO

# Name of the claims(s) to use for the user's display name.
# Can have multiple attributes listed, separated with a '|' in which
# case those values will be joined with a space.
# Example: OIDC_DISPLAY_NAME_CLAIMS=given_name|family_name
OIDC_DISPLAY_NAME_CLAIMS=name

# OAuth Client ID to access the identity provider
OIDC_CLIENT_ID=abc123

# OAuth Client Secret to access the identity provider
```

```
OIDC_CLIENT_SECRET=def456

# Issuer URL
# Must start with 'https://'
OIDC_ISSUER=https://instance.authsystem.example.com

# The "end session" (RP-initiated logout) URL to call during BookStack logout.
# By default this is false which disables RP-initiated logout.
# Setting to "true" will enable logout if found as supported by auto-discovery.
# Otherwise, this can be set as a specific URL endpoint.
OIDC_END_SESSION_ENDPOINT=false

# Enable auto-discovery of endpoints and token keys.
# As per the standard, expects the service to serve a
# `<issuer>/.well-known/openid-configuration` endpoint.
OIDC_ISSUER_DISCOVER=true

#############################################################
## NOTE: The below are only needed if not using the above ##
##       auto-discovery option.                            ##
#############################################################

# Path to identity provider token signing public RSA key
OIDC_PUBLIC_KEY=file:///keys/idp-public-key.pem

# Full URL to the OIDC authorize endpoint
OIDC_AUTH_ENDPOINT=https://instance.authsystem.example.com/v1/authorize

# Full URL to the OIDC token endpoint
OIDC_TOKEN_ENDPOINT=https://instance.authsystem.example.com/v1/token

# Full URL to the OIDC userinfo endpoint
# Won't be used if all required claims are provided in the ID token.
OIDC_USERINFO_ENDPOINT=https://instance.authsystem.example.com/v1/userinfo
```

A user in BookStack will be linked to an OIDC provided account via the `sub` claim. If the value of this ID changes in the identity provider it can be updated in BookStack, by an admin, by changing the "External Authentication ID" field on the user's profile.

Authentication System Configuration

With OIDC you don't often need to configure much specifically for BookStack. Most often, you'll just need to ensure any callback/redirect URIs are set as below:

- **Login Callback/Redirect URI:** `https://example.com/oidc/callback`
- **Logout Callback/Redirect URIs:**
  - `https://example.com`
  - `https://example.com/login`
  - `https://example.com/login?prevent_auto_init=true`
  - *Only one URL will actually be used but it depends upon specific configuration set. Some systems will allow you to instead use a wildcard like* `https://example.com/*`.

Change `https://example.com` to be the base URL of your BookStack instance.

Since v24.02 BookStack will make use of Proof Key for Code Exchange (PKCE) during authentication. If your authentication system provides the option, you should enforce PKCE to be required for extra security.

Switching to OIDC with Existing Users

When switching `AUTH_METHOD` from `standard` to `oidc`, BookStack will not link OIDC user accounts to existing BookStack users, where the email address is matching, since the "External Authentication ID" value of the existing BookStack user does not match the unique user ID provided by the OIDC system.

You can overcome this situation by logging into BookStack with an admin account while `AUTH_METHOD=standard`. While logged in, change `AUTH_METHOD` to `oidc`. This change of authentication method will show an "External Authentication ID" text field, below the name and email inputs, when viewing a user account in BookStack. Here you can enter the unique user ID that would be provided by your OIDC provider. Once saved BookStack will then use this value to match OIDC and BookStack user accounts upon next login attempt.

If you need to update accounts in bulk, you could instead directly update the `external_auth_id` field of the `users` table within your BookStack database.

Debugging

To help when setting up or configuring BookStack to use your OIDC system, the below `.env` option can help provide more insight:

```
# Dump out the details fetched from the identity provider.
# Only set this option to true if debugging since it will block logins
# and potentially show private details.
OIDC_DUMP_USER_DETAILS=false
```

Further to this, details of any BookStack errors encountered can be found by following our general debugging documentation.

Using a Different ID Claim

By default, BookStack will use the `sub` claim as a unique identifier to match up a user between BookStack and the identify provider. For the vast majority of use-cases, this is fine since this claim is part of the OIDC standard.

In some very select scenarios, you may want to use a different claim as the unique identifier. This can be done by setting an `OIDC_EXTERNAL_ID_CLAIM` option in your `.env` like shown below, where the value of the option is the name of the claim:

```
# Configure a custom ID Token claim to be used as the
# "External Authentication ID" within BookStack.
OIDC_EXTERNAL_ID_CLAIM=upn
```

Note that changing this with existing BookStack OIDC users, without changing their "External Authentication ID" values, may cause issues upon future login since their existing store ID in BookStack may no longer align.

Group Sync

BookStack has the ability to sync OIDC user groups with BookStack roles. By default this will match OIDC group names with the BookStack role display names with casing ignored. This can be overridden by via the 'External Authentication IDs' field which can be seen when editing a role while OIDC authentication is enabled. If filled, the names in this field will be used and the BookStack role display name will be ignored. You can match on multiple names by separating them with a comma. Commas can be escaped with a backslash (`\,`) if you need to map using a literal comma character.

When matching OIDC groups with role names or 'External Authentication IDs' values, BookStack will standardise the names of OIDC groups to be lower-cased and spaces will be replaced with hyphens. For example, to match a OIDC group named "United Kingdom" an 'External Authentication IDs' value of "united-kingdom" could be used.

This feature requires the OIDC server to provide a claim in the ID token with an array of group names. You'll need to specify the attribute using the `OIDC_GROUPS_CLAIM` to tell BookStack what claim it can find groups on. This value can use dot-notation to access nested properties in the ID token JSON data, an example of which can be found below.

Keep in mind you can use the `OIDC_DUMP_USER_DETAILS` option, as shown in the above debugging section to dump out claim values provided by your authentication system to help understand what is being provided by your authentication system.

Here are the settings required to be added to your `.env` file to enable group syncing:

```
# Enable OIDC group sync.
OIDC_USER_TO_GROUPS=true

# Set the attribute from which BookStack will read groups names from.
OIDC_GROUPS_CLAIM=groups

# Additional scopes to send with the authentication request.
# By default BookStack only sends the 'openid', 'profile' & 'email' scopes.
# Many platforms require specific scopes to be requested for group data.
# Multiple scopes can be added via comma separation.
OIDC_ADDITIONAL_SCOPES=groups

# Remove the user from roles that don't match OIDC groups upon login.
# Note: While this is enabled the "Default Registration Role", editable within the
# BookStack settings view, will be considered a matched role and assigned to the user.
OIDC_REMOVE_FROM_GROUPS=true
```

Nested Groups Claim Example

The below shows a reduced example of JSON data for an ID token, that has group data within a nested property, along with the `OIDC_GROUPS_CLAIM` value that would be used for this structure to detect the provided "Editor" and "Admin" roles.

```
OIDC_GROUPS_CLAIM=resource_access.bookstack.roles
```

```
{
  ...
  "resource_access": {
    "bookstack": {
      "roles": [
        "Editor",
        "Admin"
      ]
    }
  },
  "email": "email@example.com"
```

```
  ...
}
```

□□□ #4
□ cocobook-sync □□□ 2026-10-28 10:56:16 UTC
□ cocobook-sync □□□ 2026-10-29 01:42:29 UTC