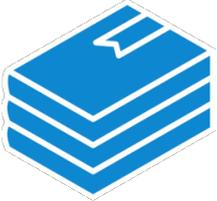# Add 3rd Party Authentication (Google, Twitch etc.)

BookStack supports third-party authentication, allowing users to log in with services like Google, GitHub, Twitter, and others. By default, these services are disabled, but you can enable them by configuring the necessary credentials from each external service.

This guide will walk you through setting up third-party authentication, focusing on the most popular services: Google, GitHub, Twitter, Facebook, Slack, AzureAD, Okta, GitLab, Twitch, and Discord.

Step 1: Enable Automatic Registration (Optional)

If you want to auto-register users when they log in via a third-party service, add the following option to your `.env` file or to your `docker-compose.yaml` :

```
{SERVICE}_AUTO_REGISTER=true
```

For example, if you're setting up Google authentication:

```
GOOGLE_AUTO_REGISTER=true
```

> This will allow users to register through third-party login services even if general registration is disabled.

Step 2: Enable Automatic Email Confirmation (Optional)

You can also enable automatic email confirmation, skipping the confirmation step for trusted third-party login services:

```
{SERVICE}_AUTO_CONFIRM_EMAIL=true
```

For example, for Google:

```
GOOGLE_AUTO_CONFIRM_EMAIL=true
```

Step 3: Configure Individual Third-Party Services

- **Google**

- **Github**

- **Twitter**

- **Facebook**

- **Slack**

- **AzueAD (Microsoft)**

- **Okta**

- **Twitch**

- **Discord**

Google Authentication

1. Open the Google Developers Console.
2. Create a new project (May have to wait a short while for it to be created).
3. In 'API and Services' go to the 'OAuth consent screen' section and enter a product name ('BookStack' or your custom set name) along with any other required details until you can save your consent screen.
4. Now in the 'API and Services' > 'Credentials' section click 'Create Credentials' > 'OAuth client ID'.
5. Choose an application type of 'Web application' and enter the following urls under 'Authorized redirect URIs', changing `https://example.com` to your own domain where BookStack is hosted:
   - `https://example.com/login/service/google/callback`
   - `https://example.com/register/service/google/callback`
6. Hit 'Create' then take note of the 'Client ID' and 'Client secret' which you'll use in the next step.
7. Add or set the following items in your `.env` file like so:

   ```
   # Replace the '{client_id}' and '{client_secret}' below with your Google Client ID and Client secret
   GOOGLE_APP_ID={client_id}
   GOOGLE_APP_SECRET={client_secret}
   ```

Users can now register and log in using their Google accounts.

GitHub Authentication

1. While logged in, open up your <u>GitHub developer applications</u>.
2. Click 'Register new application'.
3. Enter an application name ('BookStack' or your custom set name) and a link to your app instance under 'Homepage URL'. The 'Authorization callback URL' can be the root (homepage) URL for your BookStack instance. Once those details are set, select 'Register application'.
4. A 'Client ID' and a 'Client Secret' value will be shown. Add or set the following items in your BookStack `.env` file like so:

```
# Replace the '{client_id}' and '{client_secret}' below with your GitHub Client ID and Client secret
GITHUB_APP_ID={client_id}
GITHUB_APP_SECRET={client_secret}
```

> Users can now log in with their GitHub accounts.

Twitter Authentication

Before creating a Twitter application for signing in, you will need to have signed up and be approved on the <u>Twitter Developer</u> site. Part of this will require describing your use of the API.

1. Go to your <u>Twitter Developer Portal</u>, after being approved by twitter as described above. Navigate to 'Projects and Apps' > 'Overview' and under 'Standalone Apps' click 'Create App'.
2. Enter an application name and save/continue to the next step.
3. You'll now be shown some keys and tokens. Copy out the shown 'API key' and 'API secret key' values for the next step.
4. Within your BookStack `.env` file add in extra options for your token and secret like so:

```
# Replace the below '{api_key}' and '{api_secret}' with your Twitter API key and API secret
TWITTER_APP_ID={api_key}
TWITTER_APP_SECRET={api_secret}
```

Back within the Twitter developer dashboard, find your new standalone app and click on 'App Settings' then click on edit within the 'Authentication settings' section.

5. Enable the '3-legged OAuth' and 'Request email address from users' options.
6. Enter the following URLs under 'Callback URLs', changing `https://example.com` to your own domain where BookStack is hosted:
   - `https://example.com/login/service/twitter/callback`
   - `https://example.com/register/service/twitter/callback`
7. Fill in any remaining required URLs then click save.

> Users can now log in with their Twitter accounts.

Facebook Authentication

1. Navigate to the <u>Facebook developers page</u> then go 'My Apps' -> 'Add a New App'.
2. Enter an app name ('BookStack login' or something custom) and contact email then continue.
3. In your new app select 'Add Product' on the left sidebar then choose 'Facebook Login' by clicking the 'Get Started' button. Select the 'Web' option if asked to choose a platform.
4. Enter the your base BookStack url into the 'Site URL' box and save.
5. On the left sidebar again go to 'Facebook Login' -> 'Settings'.
6. Enter the following URLs under 'Valid OAuth Redirect URIs', changing `https://example.com` to your own domain where BookStack is hosted:
   - `https://example.com/login/service/facebook/callback`
   - `https://example.com/register/service/facebook/callback`
7. Navigate back to the app 'Dashboard' in the sidebar to find your app id and secret. Add or set these to your `.env` file like so:

```
# Replace the below '{app_id}' and '{app_secret}' with your Facebook app ID and secret
FACEBOOK_APP_ID={app_id}
FACEBOOK_APP_SECRET={app_secret}
```

> Users can now log in with their Facebook accounts.

Slack Authentication

1. Go to the **Slack apps page** and select 'Create An App', then 'From scratch' when prompted.
2. Enter an app name ('BookStack login' or something custom) and your workspace then select "Create App".
3. Within an "App Credentials" section, you should find your client ID and secret. Copy these details and add them as new variables in your `.env` file like so:

   ```
   # Replace the below '{client_id}' and '{client_secret}' with your Slack client ID and secret
   SLACK_APP_ID={client_id}
   SLACK_APP_SECRET={client_secret}
   ```

4. In your slack app go to 'OAuth & Permissions', find the 'Redirect URLs' section then 'Add New Redirect URL'. Enter your BookStack base URL then 'Add' before pressing 'Save URLs'.

> Users can now log in with their Slack accounts.

AzureAD (Microsoft) Authentication

> **Note:** If you intend all users to access your instance via Azure, then using an alternative primary authentication option like OIDC or SAML 2.0 will provide a better user experience while having more features like auto-login and group sync. A video guide for setting up OIDC with AzureAD can be found here.

1. Login to your your azure portal and navigate to the 'Azure Activity Directory' area.
2. Under 'Manage > App registrations' select 'New application registration'.
3. Enter a name ('BookStack'). Set the 'Redirect URI' to the "Web" platform with the value set to the following, replacing 'https://example.com/' with your base BookStack url:
   - `https://example.com/login/service/azure/callback`

4. Once created, View the application 'Overview' page and note the 'Application (client) ID' and 'Directory (tenant) ID' values. These are the APP_ID and TENANT values for step 9.
5. Within your application in azure, Navigate to 'Certificates & secrets' then choose 'New client secret'.
6. Enter any description you want and set an expiry duration. Then click 'Save'.
7. Copy the string of characters under 'Value'. This is the APP_SECRET value for step 9 and is only shown once.
8. Navigate to 'API permissions' for your app. You should already have a "Microsoft Graph" > "User.Read" permission assigned. If not choose 'Add a permission'. Find the 'Microsoft Graph' option within this, then select 'Delegated permissions' then find & select the 'User.Read' permission. Then select 'Add permissions' at the bottom of the page.
9. Copy these details and add them as new variables in your `.env` file like so:

```
# Replace the below '{APP_ID}', '{APP_SECRET}' and '{TENANT}' values with your Azure APP_ID and
APP_SECRET and TENANT
AZURE_APP_ID={APP_ID}
AZURE_APP_SECRET={APP_SECRET}
AZURE_TENANT={TENANT}
```

Users can now log in with their AzureAD accounts.

Okta Authentication

Note: If you intend all users to access your instance via Okta, then using an alternative primary authentication option like OIDC or SAML 2.0 will provide a better user experience while having more features like auto-login and group sync.

1. Login to Okta and, once logged in, Note the current URL. This is used for the 'base_url' in step 6.
2. Navigate to the Admin panel then 'Applications' then select 'Add Application'. Then select 'Create New App' on the left.
3. For the 'Platform' choose 'Web'. For the 'Sign on method' choose 'OpenID Connect' then click 'Create'.
4. Give the app a name such as 'BookStack' or 'Our documentation'. Under the 'Login redirect URIs' option add both of the below URLs, Changing `https://example.com` to the base URL of your BookStack instance:

- https://example.com/login/service/okta/callback
- https://example.com/register/service/okta/callback

5. Save and scroll down to the 'Client Credentials' area. Copy the 'Client ID' and 'Client secret' values for the next step.
6. Copy these details and add them as new variables in your `.env` file like so:

```
# Replace the below '{client_id}' and '{client_Secret}' with your Okta client ID and secret
OKTA_APP_ID={client_id}
OKTA_APP_SECRET={client_secret}
# Replace the '{base_url}' below with the URL from step 1
# but with everything after the domain (okta.com) removed.
OKTA_BASE_URL={base_url}
```

Users can now log in with their Okta accounts.

Twitch Authentication

To allow Twitch sign-in you'll first need to create an application from the Twitch developer site. Here's the process:

1. Login into the Twitch developer website.
2. Navigate to your 'Dashboard' then 'Apps' and select 'Register Your Application'.
3. Set a name to identify the application, such as 'BookStack Authentication', and in the 'OAuth Redirect URI' input add the below URL, Changing `https://example.com` to the base URL of your BookStack instance:
   - https://example.com/login/service/twitch/callback
4. Under the 'Application Category' option select 'Website Integration' then hit 'Register'.
5. Click the 'New Secret' button and accept the prompt that appears. You should now see both a 'Client ID' and 'Client Secret' value which you'll use in the next step.
6. Copy the below details and add them as new variables in your `.env` file like so:

```
# Replace the below '{client_id}' and '{client_secret}' with your Twitch client ID and secret values.
TWITCH_APP_ID={client_id}
TWITCH_APP_SECRET={client_secret}
```

> Users can now log in with their Twitch accounts.

Discord Authentication

> To allow Discord sign-in you'll first need to create an application on the Discord developer site. Here's the process:

1. Login into the **Discord developer website**.
2. Select 'Create an application'.
3. Set a name to identify the application, such as 'BookStack Authentication', and save.
4. In the sidebar, Open the OAuth2 settings for your application and add a redirect. Input the below URL, Changing `https://example.com` to be the base URL of your BookStack instance then save:
   - `https://example.com/login/service/discord/callback`
5. Back in the 'General Information' section find the 'Client ID' and 'Client Secret' values which you'll use in the next step.
6. Copy the below details and add them as new variables in your `.env` file like so:

   ```
   # Replace the below '{client_id}' and '{client_secret}' with your Discord client ID and secret values.
   DISCORD_APP_ID={client_id}
   DISCORD_APP_SECRET={client_secret}
   ```

> Users can now log in with their Discord accounts.

Revision #16
Created 8 September 2024 10:20:37 by aeoneros
Updated 9 September 2024 17:48:39 by aeoneros