Authenticating (Client Credentials)

The OIDC client credentials flow approach to authenticating API access is the simplest way to connect and authenticate your client program. It is similar to using an API key to authenticate, which you may have used when integrating your systems with other APIs.

> Client credentials flow is suitable only if the application you are integrating is a trusted client – for example, it is only installed on a server computer that you can guarantee secure access. Ensure that you understand the implications of the available OIDC flow types and when they should be used. It is your responsibility to secure your software client and the OIDC flow type that is used.

Register Application

To connect your application with this flow, you must first register your application with POS.

Navigate to the `Security / OpenID Connect / Applications` page.

> Contact Fusion Support if you do not see this menu item. Security settings may be locked down by default for most POS users.

Choose `Add an application` to register your client application. Enter the following:

- `Client ID`
  - Enter a unique ID for your application.
  - It's recommended to use a lower-case name with no spaces or punctuation; aphanumeric characters are ok.
- `Display Name`
  - Enter a friendly name for your application.
  - This is displayed only in the Applications list page in the admin UI.
- `Type`
  - Choose Confidential client.
- `Client Secret`

- Enter a long, random secret value for the application. This is like a password or API key.
- It’s recommended to use a password manager to generate a securely random value.
- Once you enter the secret, store it only in a safe place such as a password manager.

- `Flows`
  - Enable only `Allow Client Credentials Flow`.
- `Client Credentials Roles`
  - Select the user permission roles that you want to enable for the application.

> See Roles for a list of common roles, and how custom roles can be created.

Connect your Application

You can now develop your application and authenticate with POS. You will typically use a client library to handle the authentication details. The mechanism to perform authentication will depend on the client library.

Your application will need to gather the client ID and client secret that was was entered in the application definition your created above. Like any password or API key, you should ensure that these values are not hard-coded into your application, and should be referenced from secured settings (e.g. an access-controlled configuration file, or environment variables).