Companies House

Developer Hub

Sign out

Developer forum (https://forum.aws.chdev.org/)

API authentication

API and stream key authentication

The Companies House API requires authentication credentials to be sent with each request. In most cases this can be sent as an API or stream key for streaming API connections.

Sending the key

The Companies House API uses HTTP basic access authentication to send an API key or stream key between the client application and the server.

Basic authentication usually consists of a username and password. The Companies House API takes the username as the API or stream key and ignores the password, so it can be left blank.

Example of HTTP basic authentication

For an API key of `my_api_key`, the following curl request demonstrates the setting of the `Authorization` HTTP request header, as defined by RFC2617:

```
curl -XGET -u my_api_key: https://api.company-information.service.gov.uk/company/00000006
```

```
GET /company/00000006 HTTP/1.1
Host: api.company-information.service.gov.uk
Authorization: Basic bXlfYXBpX2tleTo=
```

OAuth 2.0 authentication

Interaction with some Companies House API functionality requires OAuth 2.0 authorisation. In web server apps, interaction with the Companies House API requires end-user involvement for authentication to prove their identity before the API will allow access. The Companies House API uses HTTP bearer access authentication to send an access token between the client application and the server.

Overview

In a web server process flow, there must be end user involvement. The process flow is as follows:

Developer setup

1. The developer registers an application.
2. The developer then creates a web client for their application obtaining a `client_id` and a `client_secret` which must be stored securely by the developer.
3. The web server application must be configured to use this `client_id` and `client_secret` combination for interactions with the Companies House OAuth 2.0 service.

Initiating the OAuth web server flow

1. When the web server wants to sign an end user in with their Companies House account, it redirects the user's browser to the `/oauth2/authorise` endpoint with the developer's `client_id` and other details including the requested scopes and a registered `redirect_uri`.
2. The end user signs in to their Companies House account and provides a Company authentication code if any requested scopes contain a specific company number.
3. The end user will be prompted to grant access for the application to perform certain actions on their behalf.
4. When the end user grants this permission, they will be redirected to the `redirect_uri` provided with a `code` parameter to be used in the next stage.

Handling the redirect back

- The web server can then use the code to exchange for the users `access_token` and `refresh_token` by making a `POST` request to the `/oauth/token` endpoint.
- This request is not done via a browser but directly from the web server to the Companies House OAuth 2.0 service.
- This request body includes the `code`, the developer's `client_id`, `client_secret` and some other relevant information. The `grant_type` must be set to `authorization_code` to exchange an authorization code for an access token.
- The Companies House OAuth 2.0 service verifies this request, and if access is permitted, responds with access and refresh tokens.
- The application uses this access token when making requests to the Companies House API.

Verifying the access token

- The web server can verify the access token before using it against other Companies House APIs.
- To verify the access token, the web server should make another request directly (not in the user's browser) to the `/oauth/verify` endpoint.

Refreshing an access token

- When the access token expires, the application can use the `/oauth/token` endpoint again to exchange the refresh token for a new access token. The `grant_type` must be set to `refresh_token` to exchange a refresh token for an access token.

Example of HTTP bearer authentication

For an access token of `my_access_token`, the following curl request demonstrates the setting of the `Authorization` HTTP request header, as defined by RFC2617:

```
curl -XGET -H "Authorization: Bearer my_access_token" https://api.company-
information.service.gov.uk/company/00000006
```

```
GET /company/00000006 HTTP/1.1
Host: api.company-information.service.gov.uk
Authorization: Bearer my_access_token
```

OAuth 2.0 service specifications

Each Companies House OAuth 2.0 service endpoint is documented with examples within the API specifications list in the Developer's API suite (https://developer-specs.company-information.service.gov.uk/).

Companies House has also written an example third party test harness application (https://github.com/companieshouse/third-party-test-harness) that shows how a web server application can interact with the Companies House OAuth 2.0 service. The README of this GitHub repository details how to configure and run the test harness.
