Department
for Education

DfE Developer Hub

May 2019

Developer Hub

Use this service to connect your MIS product to Department for Education (DfE) systems by API.

We are testing this service with selected users. It is not currently available to everyone.

How it works

We have developed some APIs to make it easier for schools to send and receive important data. This includes school census data.

The APIs will work with MIS applications used by schools.

This guidance is for developers and explains how to integrate an MIS application using the APIs.

We are currently testing this service with a small number of suppliers.

Before you start

You should take time to read about our open APIs. This will tell you what is currently available and how they work.

If you have an authorised DfE Sign-in account, you can also read about our restricted APIs.

Our APIs are secured using OAuth. You should read our OAuth 2.0 specification (opens in a new tab) to learn how this works**.**

Our Reference Guide section provides useful information, such as the base URLs required for accessing APIs.

How to register an application

We are testing this service with a small group of suppliers. We have given each supplier a set of instructions that explain how to register an application.

If you need technical support you should contact us on Slack.

Authorisation

Introduction

Our APIs have two types of endpoints. Each endpoint has its own:

- access level
- authorisation token
- authorisation procedure

These are dependent on the data that is served.

Endpoint access level	Required authorisation token
Open access	**No token**
User-restricted	**OAuth 2.0** `access_token`

Department for Education (DfE) APIs are only accessible over TLS (https://).

Open access endpoints

These endpoints are unrestricted and service general data we hold. They don't allow you to access secure data.

This endpoint is unrestricted so you can access it without an OAuth 2.0 access_token. It will still work if you provide a tokens as an `Authorization` header.

Example of how to call an open API

```
POST /api/{resource} HTTP/1.1
Host: https://{base_url}
Accept: application/json
```

Key	Value
resource	**The API resource to call, for example cbds**[CI1]
base_url	**The base url endpoint to call**

Most APIs will handle response types in JSON (default) or XML.

You need to replace application/json with application/xml to receive an XML response in the example shown above.

User-restricted endpoints

These endpoints require specific authorisation from the end user. They generally give access to sensitive personal data.

We use the open standard OAuth 2.0 (opens in a new tab). This allows the end user to authorise your application to connect with the Department for Education (DfE), without sharing their access credentials.

The end user authenticates directly with us using their DfE account, and authorises for specific scopes (permissions).

We then issue an OAuth 2.0 access token (opens in a new tab) which is specific to the end user. Your application passes the access token to user-restricted endpoints in subsequent API requests.

The access token gives access to the end user's data.

Authorisation rules for specific API endpoints are given in the API documentation.

The access token expires after a period of time and must be refreshed. The expiry time will be shown in the response back when you receive the token. Your application must check the expiry before sending a request with the access token.

After 14 days you can no longer refresh the access token and the end user must authorise your application again.

Getting an OAuth 2.0 access token

The authorisation user journey is an important part of our security. It is also subject to change without notice.

! You must not modify this journey in any way.

End users must give consent before a supplier can send their data. We use the OAUTH authorization code grant flow (opens in a new tab). To give consent you must call the authorise endpoint on our Send Data to DfE service.

```
GET https://{ authorisation_base_url}/auth?response_type=code
&client_id={client_id}
&redirect_uri={redirect_url}
&scope=openid+profile+email+organisation+offline_access
&prompt=consent
&role_scope=School+Census+Summer+2019
&state={state}
```

! Do not include your client secret in this request.

Parameter	Description
response_type	The OAuth 2.0 response type. Currently the only acceptable value is `code`.
client_id	The client id associated with the MIS supplier application. Available from the DfE Developer Hub on registering your application.

redirect_uri	The URL that we use to send the authorisation results back to your application after successful (or unsuccessful) authorisation. This must match one of the redirect URLs you specified when you created your application. The redirect_uri must be named is `redirect_uri` in the call otherwise the authorisation code exchange will fail.
scope	A space-delimited list of scopes you would like to have permission to access on behalf of your user. Must be percent-encoded, so spaces must be represented as either `%20` or `+`.
prompt	Specifies whether the authorisation server prompts the End-User for consent. Currently the only acceptable value is `consent`.
role_scope	Currently the only valid value is `School+Census+Summer+2019`, this is displayed on the consent page to the end user.
state (optional)	An opaque value used to maintain state between the request and callback and to prevent tampering as described in the OAuth 2.0 specification. This is passed back to your application via the redirect_url.

Receive authorisation results

You need to create an endpoint in your application to receive the authorisation results, which needs to support an HTTP GET to the redirect URL you specified.

We'll redirect the user's browser back to your endpoint once the user has granted your application the requested authority.

Example of the authorisation request:

```
GET https://{redirect_url}?code{authorisation_code}
&state={state}
&session_state={session_state}
```

If the user does not consent, you will receive this response:

```
GET https://{redirect_url}?error=consent_denied
```

Exchange authorisation code for access token

When you get the authorisation code, you must exchange this for an access token **within 10 minutes** before it expires.

Do this via a POST to our token endpoint.

Include the request parameters in the request body as a URL-encoded string, not as request headers.

Example of a request to exchange authorisation code

```
POST https://{authorisation_base_url}/token
Headers:
Content-Type application/x-www-form-urlencoded
Authorization Basic ABC4dGVzdDNvbnNlbnRjaGlsZD…
```

```
Request body:
grant_type authorization_code
redirect_uri {redirect_url}
code {authorization_code}
```

Use the HTTP Basic authentication scheme as defined in [RFC2617] to authenticate with the authorization server. The client identifier is encoded using the "application/x-www-form-urlencoded". The client sends the client_id and client_secret, separated by a single colon (":") character, within a base64 encoded string in the credentials.

Parameter	Description
grant_type	The OAuth 2.0 grant type. Currently the only acceptable value is `authorization_code`
redirect_uri	The same redirect URL you used to call the authorisation endpoint.
code	The authorisation code you received from us in the previous step.

The response contains the access token used for calling the APIs and a refresh token used to obtain a new access token once the current one expires.

Example of a successful exchange response:

{
  "access_token":"YWE3MmQyZDAtOWQyZC00M2I2LWIxZWItYmVk...",
  "expires_in":3600,

```
  "id_token":"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZ...",
  "refresh_token":"MmZmNTMzMGYtZGRhYi00MDI1LWFiNWUtZjc2...",
  "scope":"openid profile email organisation offline_access",
  "token_type":"Bearer"
}
```


How to refresh an access token

When the access token expires you will need to send a POST request to exchange a refresh token for a new access token. The response will include a new refresh token that will replace the previous one and must be retained and used the next time the access token expires.

You should send the POST request to:

```
POST https://{authorisation_base_url}/token
Headers:
Content-Type application/x-www-form-urlencoded
Authorization Basic ABC4dGVzdDNvbnNlbnRjaGlsZD…
Request body:
grant_type refresh_token
refresh_token {refresh_token}
```

Example of a successful token response


```
{
  "access_token":"ZmQ5ODcwYTYtM2U2My00OWU0LWJjYzktZDljZjc...",
  "expires_in":3600,
  "id_token":"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCIsImtpZCI...",
  "refresh_token":"YWIwNjQ0NTUtMWVmZC00NWJjLThlMjItNmZhYjc2MGQ0NzI4QN6pGm...",
  "scope":"openid profile email organisation offline_access",
  "token_type":"Bearer"
}
```


! You must store your access and refresh tokens securely and not share them with anyone else.

Calling a user-restricted API

To call one of the restricted APIs requires the use of the access token. In addition, some APIs may require a subscription key that can be obtained from the Developer Hub.

Example of a call to a user-restricted API

```
POST /api/{resource} HTTP/1.1
Host: https://{base_url}
Authorization: Bearer {bearerToken}
Ocp-Apim-Subscription-Key {subscriptionKey}
```

Parameter	Description
bearer_token	The access token that is used to grant access to user-restricted APIs. The access token expires after an hour and you should check the expiry time before calling the user-restricted API. You will need to generate a new one using the refresh token if it has expired. Bearer tokens are issued as access tokens of ‘bearer’ type.
subscription_key	The subscription key provided within the DfE Developer Hub.

Reference Guide

API access

The base URL for test APIs is:

```
https:// put url to be used in MIS test here
```

The base URL for production APIs is:

```
https:// put url to be used in prod here
```

Authorization Server access

The base URL for test Authorization Server is:

```
https:// put url to be used in MIS test here
```

The base URL for production Authorization Server is:

```
https:// put url to be used in prod here
```