Aerostudies OAuth API Guide

Preface

The API as implemented by Aerostudies is RESTful and adheres to the OAuth 2.0 specifications. This particular guide is meant as a brief overview on how the OAuth 2.0 system works specifically with the Ascent system.

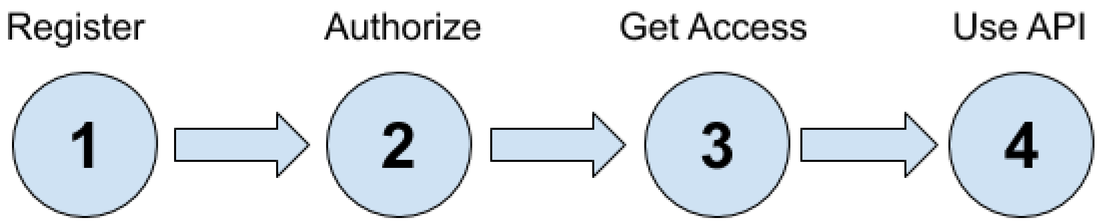
Note that for the purposes of app to app API access, Ascent only implements the authorization code workflow of the OAuth 2.0 spec.

Getting Started

PROCESS OUTLINE

In order to begin consuming and publishing data to the API, you will need to register your application with the system and have an authenticated Aerostudies user authorize your application to grant access to their LMS data.

The simplified process, broken into stages is shown below:



The API implements the OAuth 2 protocol and as such your application will have to follow the standard process outlined above which is the same process that the OAuth 2 protocol stipulates.

PRE-REQUISITES

Before you begin registering your application and going through the auth and token access as highlighted above you will need to obtain the API Id from an authorized user of the Ascent system. The API Id is a 32 character alphanumeric string, which can be found in the Settings area of the Ascent system by an authorized user with Registrar role.

Apart from the API Id, your application will also need an endpoint configured in order to receive callback data. This endpoint URI will need to be passed in the registration process in order to facilitate the OAuth 2 authorization.

REGISTER

When you have retrieved the API Id and setup a callback endpoint on your web application, you can register your app on the Ascent system. In order to do this, you will POST the required following parameters to [url]/oauth/register:

- apiId (string) [32]: The API Id from authorized users learning center
- companyName (string): Your companies name
- applicationName (string): The name of your web application
- redirect_uri (string): The URI our system will pass data back to in the auth and other processes
- callbackUri * (string, not required): The URI our system will use as a callback for any API notifications for events (Such as training completion, module status change, etc.)

- callbackEnabled * (int 0/1 or boolean, not required): Determines whether or not you want to receive API notifications at the specified callback URI. Default is false (or off).

If the register call is successful you will retrieve back a JSON object which looks like the following:

```
{
  "Client": {
    "apiId": "Qk0jmg8kC9MieGz8td8HZ3H4bpKzpU0f",
    "companyName": "Test Company",
    "applicationName": "Test Application",
    "redirect_uri": "https://testcompany.com/authRedir",
    “callbackUri”: “https://testcompany.com/callback”,
    “callbackEnabled”: true,
    "client_id": "NWQwYmUyMmYwMTViNGFh",
    "client_secret": "2d47fd95032140745bdff28a4e848080"
  }
}
```

The “client_secret” is crucial to obtaining an access token and cannot be retrieved if lost. Make sure it is securely stored as this is used to generate access tokens for API access.

AUTHORIZE

In order to authorize your web app, you will need to send an authorized user to a designated Ascent URI with specific GET parameters. The user will then sign into the Ascent system and either allow or deny your app's API access.

The URI can be built as a button or link, following the format below replacing the client_id with the returned client_id info passed from the register call as well as the redirect_uri you registered with:

[url]/oauth/authorize?client_id={**client_id**}&redirect_uri={redirect_uri}

The user will be sent to sign in, and then presented with a page that will prompt them to allow or deny your applications API access to their data.

Following a successful authorization, the Ascent API will redirect the user back to the redirect URI you specified in the registration process, and will pass along a code in the GET parameters which you will need in order to obtain an access token.

GET ACCESS

At this point your app has been allowed access from an authorized user of the Ascent system and you only need to get an access token before being able to use the API. In order to obtain an access token you will POST the following data to [url]/oauth/token:

- grant_type: "authorization_code"
- code: {**code**}
- client_id: {client_id}

- client_secret : {**client_secret**}

In return, as long as the data is correct and the “**code”** has not expired (Code expires in 5 minutes) you will be granted an access token which will stay valid for 30 days, after which you can request a refresh token.

The following is the data that is returned, and it is critical that you securely store both the **access_token** and the **refresh_token**. You will use the access_token for all API calls, and the refresh_token in order to refresh your access token.

```
{
   "access_token": "2796d3ecb6929a652db8e0ef13b60639692c3543",
   "expires_in": 2592000,
   "token_type": "bearer",
   "scope": null,
   "refresh_token": "52c09baab1775a9713de103b8333826d80ef892b"
}
```

USE API

When you have obtained an access token, for all subsequent API calls, you need to provide the access_token, client_id and client_secret in your POST or GET requests.

As an example, you may want to list all of the express modules in an LC. In this case, you would POST the following data to https://ascent.aerostudies.com/api/express/listModules:

```
{
        access_token: “2796d3ecb6929a652db8e0ef13b60639692c3543”,
        client_id: “NWQwYmUyMmYwMTViNGFh”,
        client_secret: “2d47fd95032140745bdff28a4e848080”,
}
```

In this case, listModules does not take any extra parameters, so you are just sending the access token along with the requisite auth parameters you need to send with every request.