

# CREATING WEB CONNECTIONS

## USING OAUTH 2.0 AUTHENTICATION

This step-by-step guide outlines how to create a web service connection in FME, then upload it to FME Hub using Google+ as an example.

By: Rosie Rutherford-Stone



**SAFE SOFTWARE™**

# STEP 1

## REGISTER FOR AN ACCOUNT

If you haven't done so already, create an account for the web service you want to create a connection for. In many cases like Google+, a basic account provides sufficient access to developer options to create a connection. However, in some instances, a developer account is required. Check the web service's documentation to determine what is required.

# STEP 2

## REGISTER YOUR APP

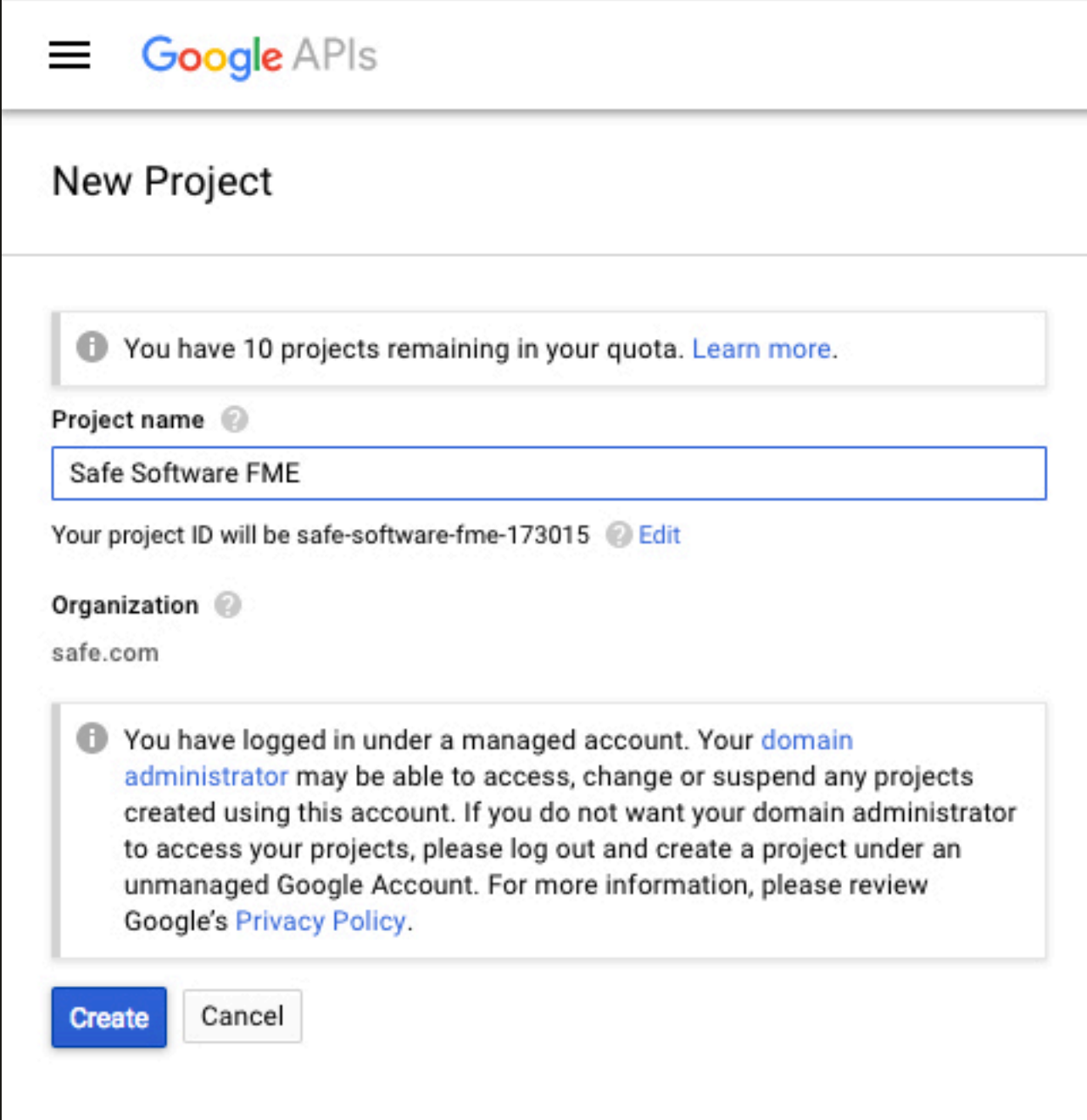
Before using OAuth with FME, you must register your application with the service.

The developer or API section of the service's website will likely have a place for you to create a new app. Here, you usually enter the following information about your app:

- Application Name: Usually in the format of <Company Name> <Service Name>
- Application Description: A brief overview of the service you are providing.
- Application Website
- Redirect URI or Callback URI: Where the service will redirect the user after they authorize (or deny) your application. Set it to <https://localhost> for FME Workbench.

# GOOGLE+ CONFIGURATION

To do this for Google+, go to the [Developers Console Library](#) and select “Create a project”. When the dialog pops up, enter “Safe Software FME” as the project, and leave the organization as the default “safe.com”. Click the “Create” button and you’ve made your new app.



The screenshot shows the 'New Project' dialog in the Google APIs console. At the top, there's a hamburger menu icon and the 'Google APIs' logo. The title 'New Project' is centered. Below the title, a message box states: 'You have 10 projects remaining in your quota. [Learn more.](#)'. The 'Project name' field is labeled with a question mark icon and contains the text 'Safe Software FME'. Below this, it says 'Your project ID will be safe-software-fme-173015' with an 'Edit' link. The 'Organization' field is labeled with a question mark icon and shows 'safe.com'. At the bottom, another message box states: 'You have logged in under a managed account. Your [domain administrator](#) may be able to access, change or suspend any projects created using this account. If you do not want your domain administrator to access your projects, please log out and create a project under an unmanaged Google Account. For more information, please review Google's [Privacy Policy](#).' At the very bottom are two buttons: 'Create' (blue) and 'Cancel' (white).

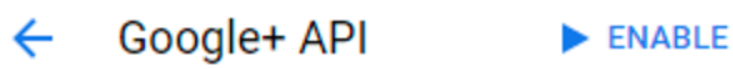


## STEP 3 : OBTAIN CLIENT ID AND SECRET

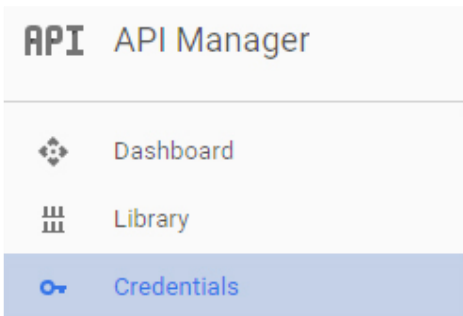
Once the app has been registered, a client ID and secret need to be generated. These are used to authenticate the identity of FME to the service API where FME requests to access a user's account. These are usually created automatically when the app is made, but sometimes need to be generated at the user's request. In most cases, there will be a link on your app's page to create this. Some web services also need you to specify the scope you are going to be using here. For the specifics of the web service you are trying to implement, check its documentation. Keep the client ID and secret on hand, as they will be needed for the next step.

# GOOGLE+ CONFIGURATION

Before doing this step for your Google+ web connection, you need to enable the Google+ API within the Developers Console. In the Library Section, click the "[Google+ API](#)" link under the Social APIs category. Next to the title "Google+ API" there is a blue arrow pointing to the "Enable" button. Clicking this allows Google+ to be accessed using your credentials, which you'll create in an upcoming step.



Next, go to the Credentials section of the API manager to configure the consent screen. This is the screen that users will see when they log into Google+ through your application. To configure this, go to the OAuth consent screen tab, specify the product name you want users to see, and privacy policy URL for your application. You can also add the optional fields of homepage URL, product logo URL and terms of service URL.



After enabling and configuring the consent screen for the API, you can create the credentials for the web connection. Since you're working with OAuth 2.0, select "OAuth client ID" from the drop-down list.

API

API Manager

Dashboard

Library

Credentials

Credentials

OAuth consent screen

Domain verification

Email address

rosie.stone@safe.com

Product name shown to users


FME

Homepage URL (Optional)

https://www.safe.com/

Product logo URL (Optional)

http://www.safe.com/inc/images/logos/fme-100x69.png



This is how your logo will look to end users  
Max size: 120x120 px

Privacy policy URL

Optional until you deploy your app

http://www.safe.com/privacy/

Terms of service URL (Optional)

http://www.safe.com/legal/

Save

Cancel



The consent screen will be shown to users whenever you request access to their private data using your client ID. It will be shown for all applications registered in this project.

You must provide an email address and product name for OAuth to work.

ASK QUESTIONS AND GET ANSWERS FROM OUR BRILLIANT COMMUNITY OF CUSTOMERS & PARTNERS IN THE [FME KNOWLEDGE CENTER](#)

Since you'll be using this to connect to FME, you need a client ID for the application type "Web Application". Here, specify the name of the web application and the redirect URIs (<https://localhost>).

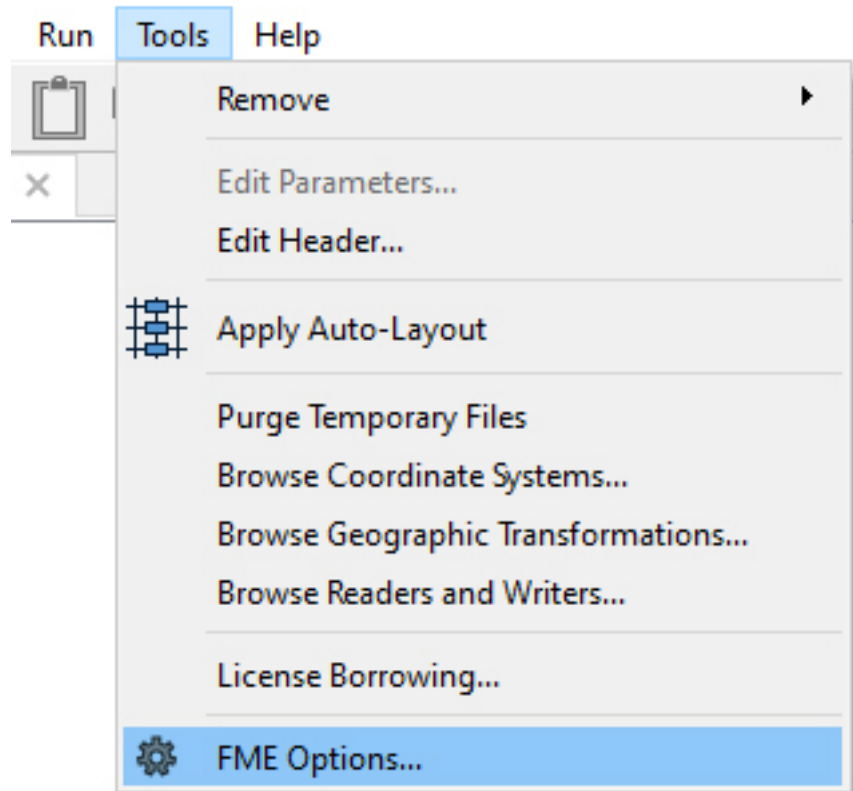
You now have your client id and secret, so you can start working with Google+ in FME.



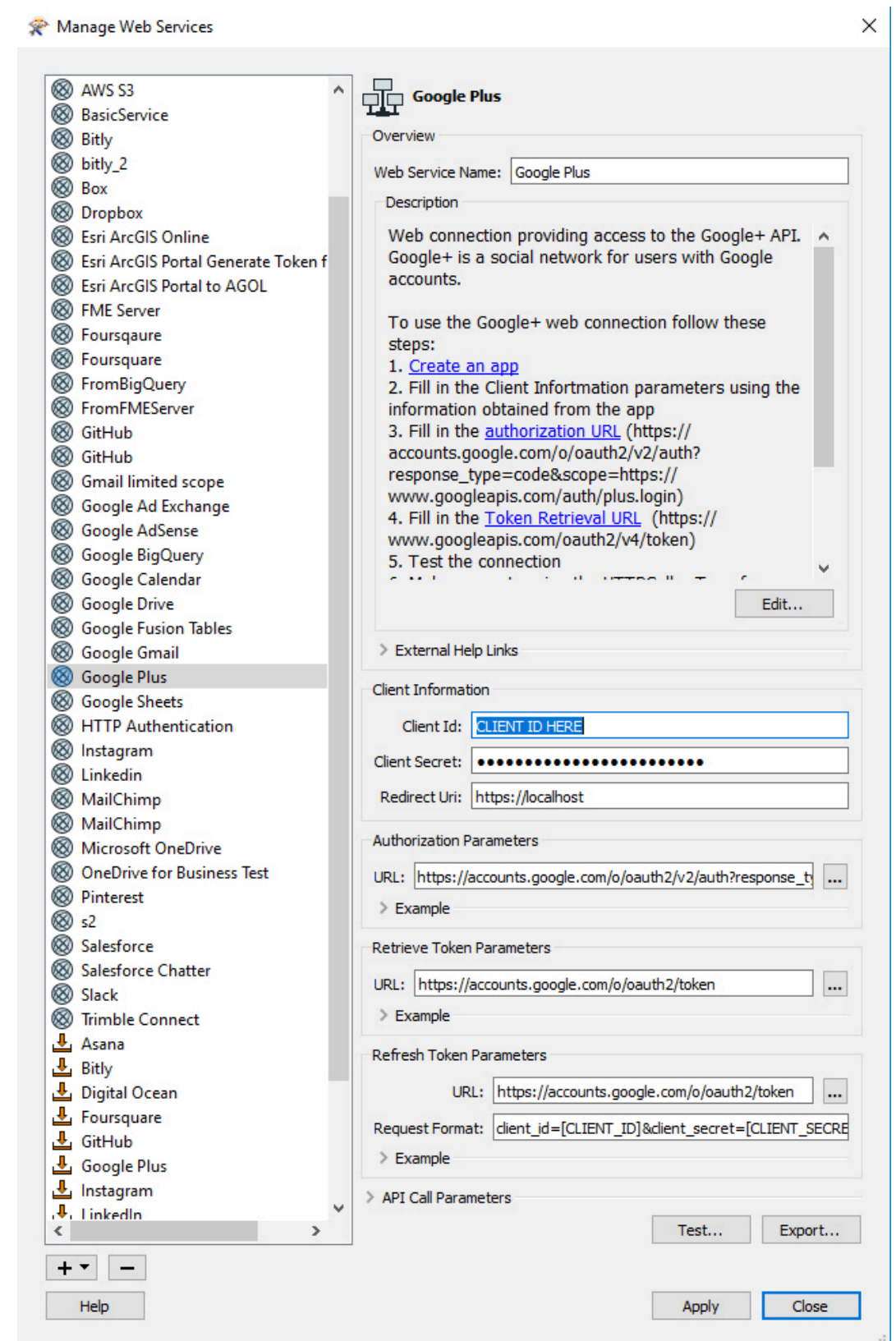
API	API Manager	← Create client ID
<div>Dashboard</div> <div>Library</div> <div>Credentials</div>		<div>Application type</div> <div><input checked="" type="radio"/> Web application</div> <div><input type="radio"/> Android <a href="#">Learn more</a></div> <div><input type="radio"/> Chrome App <a href="#">Learn more</a></div> <div><input type="radio"/> iOS <a href="#">Learn more</a></div> <div><input type="radio"/> PlayStation 4</div> <div><input type="radio"/> Other</div> <div>Name</div> <div>Safe Software FME</div> <div>Restrictions</div> <div>Enter JavaScript origins, redirect URIs, or both</div> <div>Authorized JavaScript origins</div> <div>For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (http://*.example.com) or a path (http://example.com/subdir). If you're using a nonstandard port, you must include it in the origin URI.</div> <div>http://www.example.com</div> <div>Authorized redirect URIs</div> <div>For use with requests from a web server. This is the path in your application that users are redirected to after they have authenticated with Google. The path will be appended with the authorization code for access. Must have a protocol. Cannot contain URL fragments or relative paths. Cannot be a public IP address.</div> <div>http://www.example.com/oauth2callback</div> <div>Create</div> <div>Cancel</div>

# STEP 4. CREATE A NEW WEB CONNECTION IN FME

To create a web connection, open FME Workbench and navigate to the Tools menu. From the drop-down menu, click "FME Options...". (Mac: FME Workbench > Preferences)



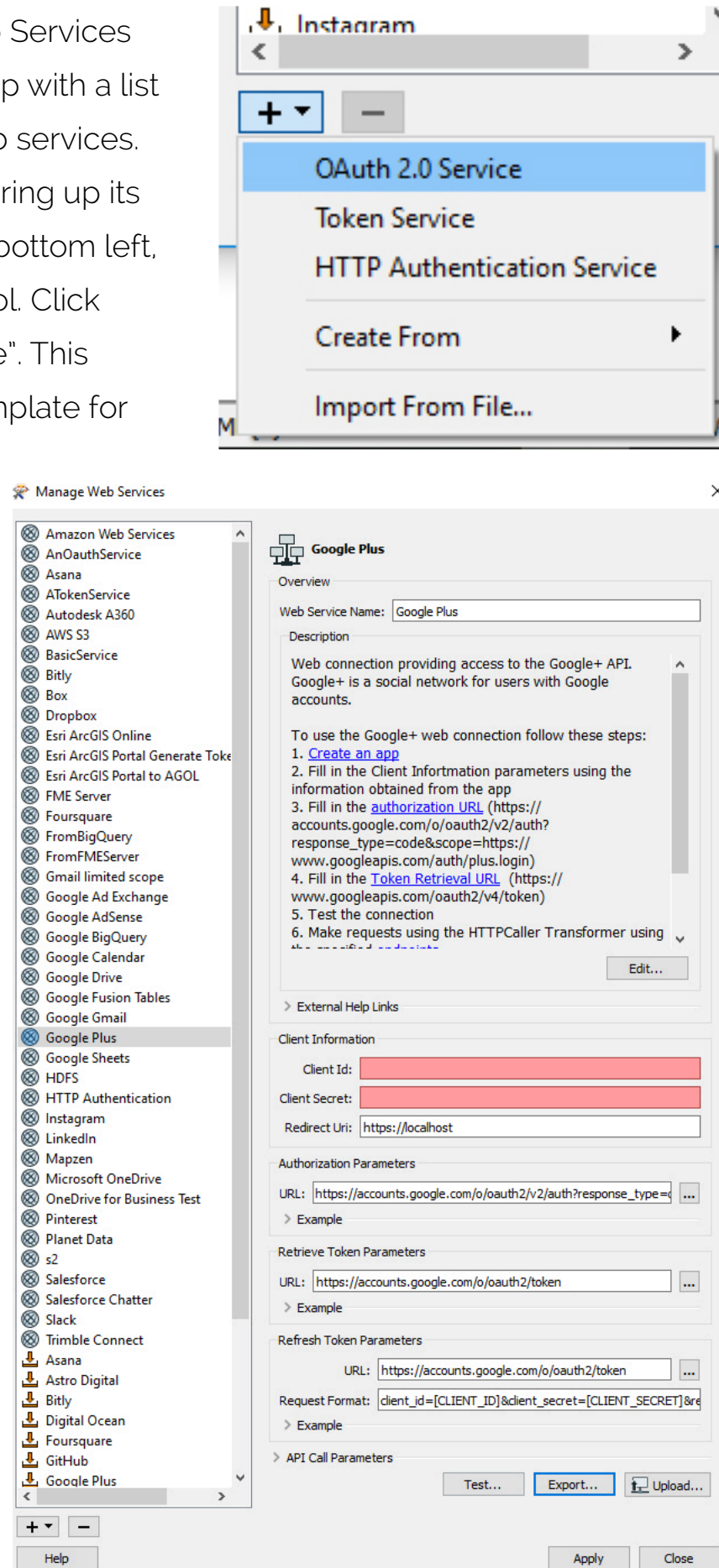
This will bring up a window with a list of options on the left side. Click "Web Connections". At the bottom right, click "Manage Services...".





The Manage Web Services window will pop up with a list of the current web services. Clicking one will bring up its definition. On the bottom left, click the "+" symbol. Click "OAuth 2.0 Service". This creates a new template for an OAuth 2.0 web service.

Give your web service a name and description. Fill out the client information section with the client ID and secret that were generated earlier, and the redirect URI.



## AUTHORIZATION PARAMETERS

The first step in obtaining access to a web service via OAuth 2.0 is to get authorization from the user. In FME this is accomplished by displaying an interface provided by the service to the user.

The "Authorization Parameters" section consists of the URL needed to authenticate the user. To find this, look in the developer documentation for the web service. It will likely be in a section titled "Authentication" or "OAuth 2.0". If it isn't obvious, check the "Getting Started" section, or use the search feature to locate it. Usually the URL is enough but occasionally more parameters such as scope and response type must be specified. This is done by appending it at the end of the URL.

## GOOGLE+ CONFIGURATION

Google+ requires a [scope](#) and response type to be specified. Be careful to set the scope up correctly otherwise the requests will not work. The information for Google+ is found on the [Using OAuth 2.0 for Web Server Applications](#) page. The authorization needs the scope and response type appended to the URL as follows:

**[HTTPS://ACCOUNTS.GOOGLE.COM/O/OAUTH2/V2/AUTH?RESPONSE\\_TYPE=CODE&SCOPE=HTTPS://WWW.GOOGLEAPIS.COM/AUTH/PLUS.LOGIN](https://accounts.google.com/o/oauth2/v2/auth?response_type=code&scope=https://www.googleapis.com/auth/plus.login)**

You can choose the scope that allows access to the user's age range, language, and a list of people in their circles.

## RETRIEVE AND REFRESH TOKEN PARAMETERS

If the user authorizes FME in the first step, the service redirects the user-agent to the application redirect URL (<https://localhost>), along with the authorization code. FME can then request an access token from the API by passing the authentication code along with authentication details.

The “Retrieve Token Parameters” section allows you to configure this and requires the API token endpoint. To find the URL, look in the documentation for anything regarding obtaining an access token.

The end result of this step is obtaining an access token that you can use to make requests to the API. FME may use the token to access the user’s account via the service API, limited to the scope agreed by the user, until the token expires or is revoked.

You also need to configure the “Refresh Token Parameters” which consists of URL and Request Format fields. The URL might be the same as the Retrieve Token URL or there may

be a dedicated refresh workflow. If there is a dedicated refresh workflow, then a Refresh Token will have been included when the original access token was issued, which FME uses to obtain a new access token. It is crucial you get this right if you are publishing the workflow to FME Server. The Request Format field allows you to indicate the desired Request Format. In most cases, the default parameters will work. This field can also be left blank.



**HOME USE LICENSE FOR FME DESKTOP**  
Learn how you can get a FREE license of FME Desktop to use outside of the workplace.

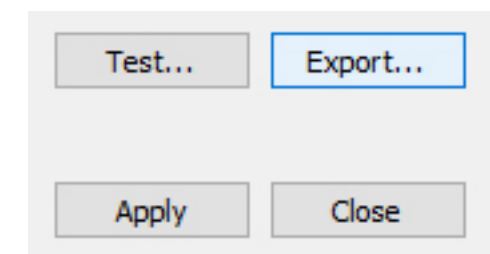
**ABOUT**  
If you wish to use FME at home to create proof of concepts, further your education, or simply just to have some fun – then you’re eligible for a free license of FME Desktop.

**APPLY**  
Fill out [this form](#) to apply for your free license. Please note that this home use license cannot be used for any production or commercial purpose.

## GOOGLE+ CONFIGURATION

Find the token information for Google+ on the same page as the [authorization](#) URL. The Retrieve Token URL and the Refresh Token URL are the same, and the default Request Format applies.

You are now ready to test your connection. Click “Apply” (Mac: “Save”) on the “Manage Web Services” dialog. Click the “Test...” button above it. Any error messages will be printed in the FME Workbench translation log pane (if this pane is hidden, turn it on under View > Windows).

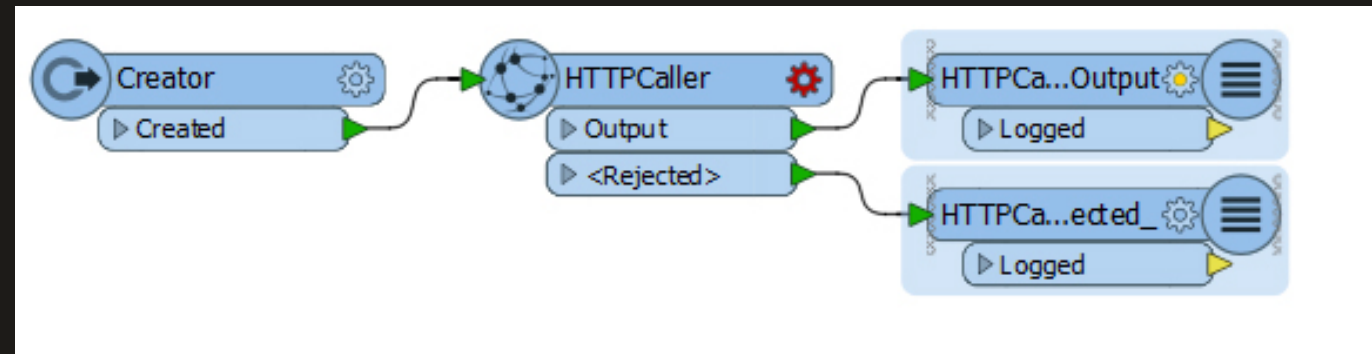


Once you have successfully connected to the web service, you are ready to test the web connection using the HTTPCaller transformer.

# STEP 5

## TEST THE CONNECTION USING THE HTTPCALLER TRANSFORMER

On the FME Workbench canvas, add a Creator and an HTTPCaller transformer. Connect them as indicated in the right image. Right-click on the HTTPCaller and click "Connect Loggers".



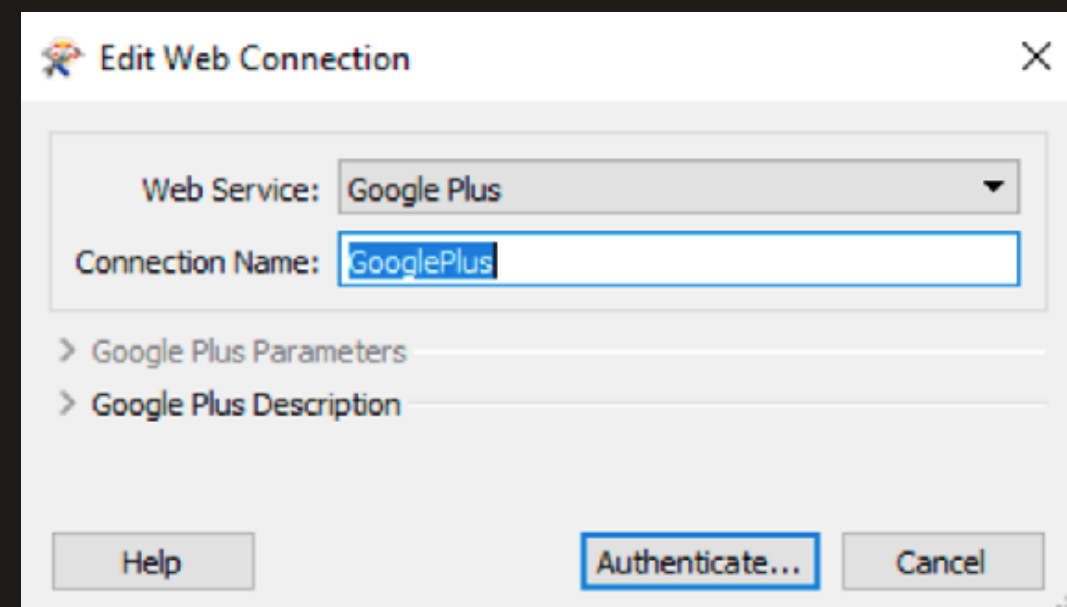
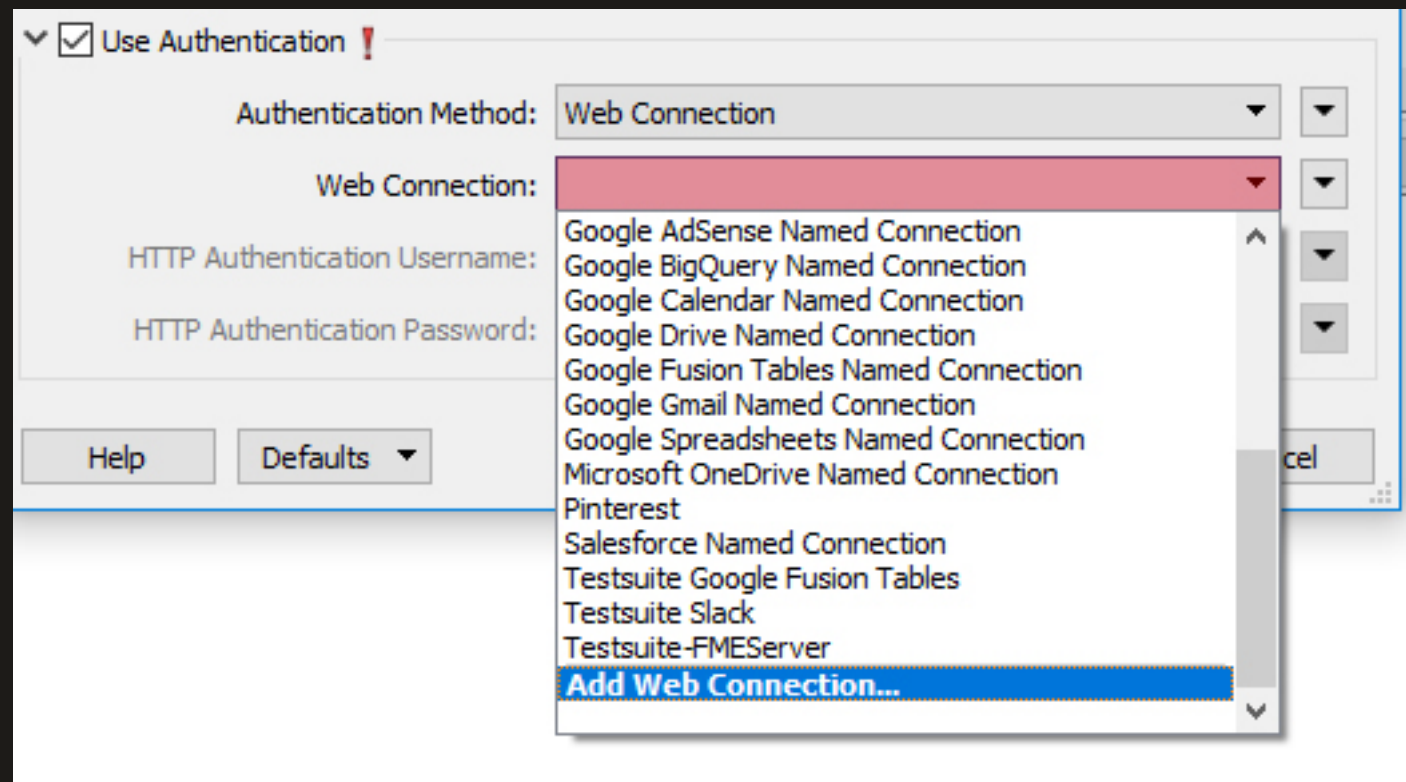
Find an endpoint in the documentation of the API to test. Open the HTTPCaller parameters and set the HTTP method to the method required to perform the request (GET, POST, etc). Make sure you know what this request URL and method should return so you can verify it works.

Find endpoints for Google+ in their [API reference documentation](#). Use the People endpoint to get your own profile information:

The screenshot shows the 'Request' tab of the HTTPCaller transformer's configuration window. It includes a 'Request URL' field with the value 'https://www.googleapis.com/plus/v1/people/me/' and a dropdown arrow. Below it is an 'HTTP Method' dropdown menu set to 'GET'. There are also expandable sections for 'Query String Parameters', 'Headers', and 'Body', each with a right-pointing arrow.

At the bottom, check the “Use Authentication” box. This brings up more options. Change “Authentication Method” to “Web Connection”. In the “Web Connection” field, scroll to the bottom of drop-down list and select “Add Web Connection...”.

A new dialog appears with two inputs: “Web Service” and “Connection Name”. Select your new web service from the drop-down list. Set the Connection Name to whatever you like. Create the web connection by clicking “Authenticate” and signing into your account.





You are now ready to test your web service by running the workspace. If you are planning to contribute your web connection to FME Hub, save this workspace as your test workspace (discussed further in step 6).

Check the translation log to make sure there are no errors. Check the output to make sure it is what you expected.

You should have something similar to this:

```
Attribute(encoded: UTF-8) : '_response_body' has value '{
  "kind": "plus#person",
  "etag": "\"FT7X6cYw9BSnPtIywEFNNGVVdio/VZjxiofX7jFR12MYCmra5pH0Twk\"",
  "gender": "female",
  "objectType": "person",
  "id": "111080890342615524092",
  "displayName": "Rosie R. Stone",
  "name": {
    "familyName": "R. Stone",
    "givenName": "Rosie"
  },
  "..."
}
```

## CODERS ON COUCHES DRINKING COFFEE



In [this episode](#), we're joined by Don Murray and Jennifer Luther Thomas, who connected littleBits to FME Server to make custom notification workflows.



# STEP 6: UPLOAD A WEB CONNECTION TO FME HUB

To share your web connection with others in the FME Community, head to [FME Hub](#). Accounts are free and give you access to hundreds of transformers, formats, templates, and web connections created by other users.

To upload a web connection to FME Hub, start in FME Workbench. First make sure you have entered a good description of what the web connection is and if there are any key documentation links add them in too. Next, erase the Client Id and Client Secret from the Client Information section. These fields will appear red as they are required, this is fine for now but you will not be able to save the web connection.

The screenshot shows the 'Manage Web Services' dialog box in FME Workbench. The left pane lists various web services, with 'Google Plus' selected. The right pane displays the configuration for the 'Google Plus' service.

**Google Plus**

**Overview**

Web Service Name:

Description

Web connection providing access to the Google+ API. Google+ is a social network for users with Google accounts.

To use the Google+ web connection follow these steps:

1. [Create an app](#)
2. Fill in the Client Information parameters using the information obtained from the app
3. Fill in the [authorization URL](#) ([https://accounts.google.com/o/oauth2/v2/auth?response\\_type=code&scope=https://www.googleapis.com/auth/plus.login](https://accounts.google.com/o/oauth2/v2/auth?response_type=code&scope=https://www.googleapis.com/auth/plus.login))
4. Fill in the [Token Retrieval URL](#) (<https://www.googleapis.com/oauth2/v4/token>)
5. Test the connection

> External Help Links

**Client Information**

Client Id:

Client Secret:

Redirect Uri:

**Authorization Parameters**

URL:  ...

> Example

**Retrieve Token Parameters**

URL:  ...

> Example

**Refresh Token Parameters**

URL:  ...

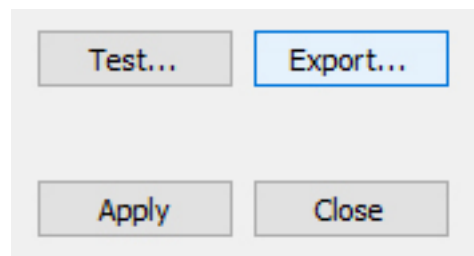
Request Format:

> Example

> API Call Parameters

In the “Manage Web Services” dialog, to the right of the “Test” button is an “Export” button. It will ask you if you want to save the changes and continue with the export. When you click “yes”, it will give you an error, telling you that the Web Service was not saved. Click “ok” on the error, and then specify the folder to export the web connection to. When closing the “Manage Web Services” dialog, it will ask again if you want to save the modifications, click “discard” and your Client Id and Secret will be restored.

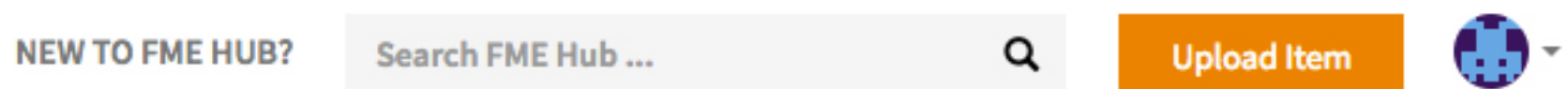
Click this and specify an output folder.



Save your test workspace from earlier — you will use it later. If you have already deleted it, you can make a new one. The only guideline is that you should be able to make a request to your web service through a REST call. The simplest way to do this would be to create a workspace similar to the one earlier using the HTTPCaller.

Open FME Hub in your browser and log in. Ensure a connection to the service does not already exist. You can filter the list of FME Hub items to Web Connections by checking the box on the left.

If your web connection does not already exist, click the orange “Upload Item” button next to the search bar at the top of the page.





You will be brought to the “Step 1: Upload Item” page. Click “Choose file to upload” and select the XML file you exported earlier. As long as you have not changed any of the tags, it will automatically detect it is a web connection.

Click “Next” and your web connection will upload.

Configure the web connection by adding a logo. The description is extracted from what you defined in FME Workbench.

Step 1  
Upload Item

Step 2  
Configure Item

Step 3  
Test Workspace

Transformer  
(.FMX)

Template  
(.FMWT)

Web Connection  
(.XML)

Format  
(.FDS)

Drag Transformer, Template, Format or Web Connection Here  
Or, if you prefer

Choose file to upload

Maximum Upload File Size 10MB

Visibility: ☒ Public  
Everyone on this site can see this item

☐ Private  
Only you can see this item (until you make it public)

Next

Upload the test workspace you saved earlier. This will make it easier to maintain the item as you will receive a notification if the test fails which could mean the web connection has stopped working due to a change in the API.

Once a web connection is created, you can edit or remove it at anytime while logged in. Go to the web connection's page and click the pencil menu to bring up the options.

★ Star | 0

Download

Make private

Edit

Remove