

Google AdWords API Provider Setup

Last Modified on 12/30/2019 4:20 pm EST

Before you can authenticate an instance of the Google AdWords connector, you must have a Google Ads manager account with access to the AdWords API and a developer token; see [Google's documentation](#) for additional information.

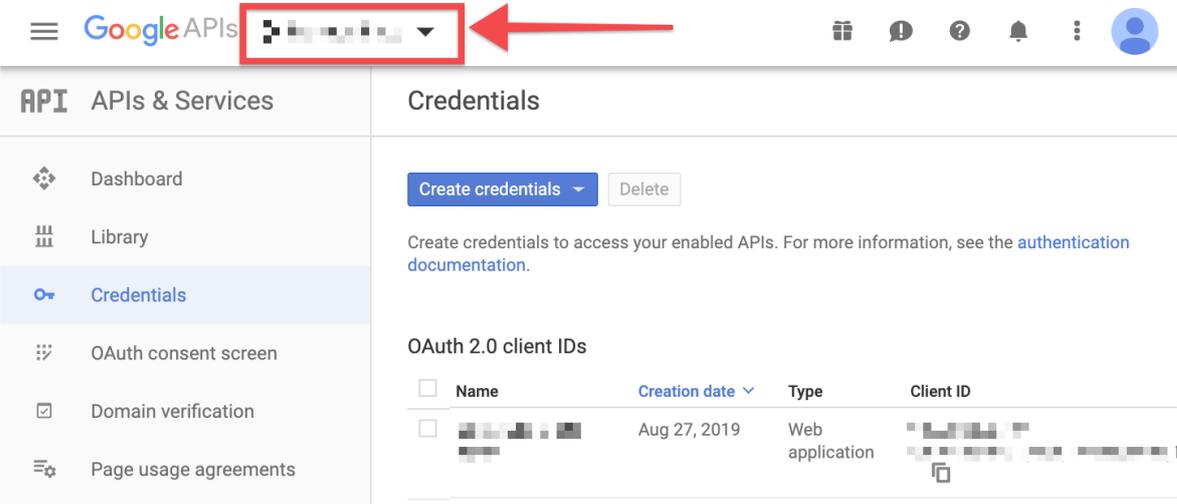
After receiving your developer token, you must also complete these steps before you can authenticate:

- [Create and register an Adwords project](#)
- [Generate OAuth credentials](#)
- [Enable the Google Ads API](#)
- [Configure the Consent Screen](#)
- [Identify the client customer ID](#)

Note: depending on which if any of these you may have already completed for other projects, some of the steps may need to be performed in a slightly different order, or may already be complete.

Creating and Registering an AdWords Project

1. Navigate to the [Google APIs Console](#) and sign in to your account.
2. Click the Project selection dropdown, then click **NEW PROJECT**.



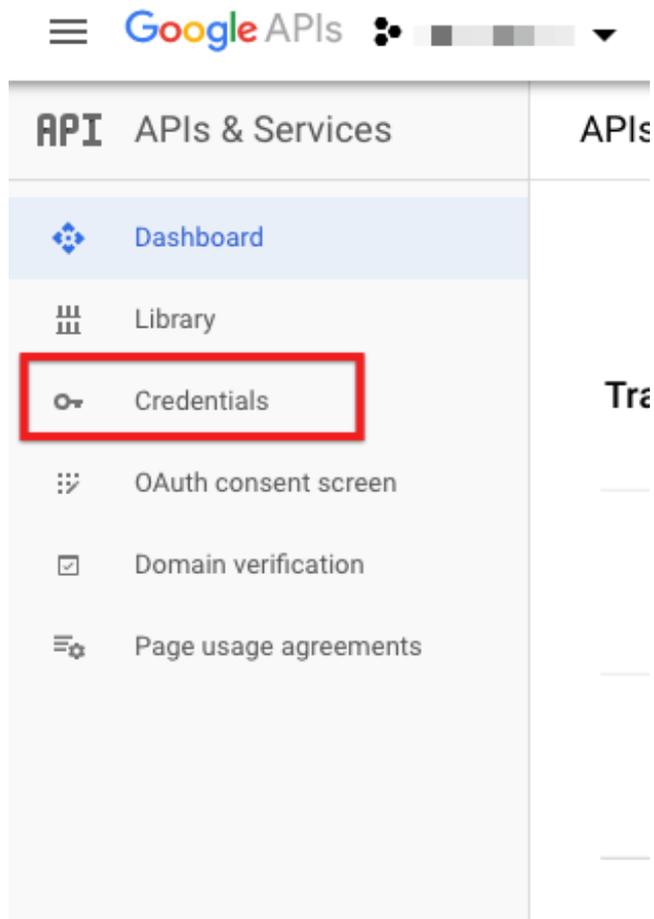
The screenshot shows the Google APIs Console interface. At the top, the 'Google APIs' logo is followed by a project selection dropdown menu, which is highlighted with a red box and a red arrow pointing to it. Below the header, the left sidebar contains a navigation menu with options: Dashboard, Library, Credentials (selected), OAuth consent screen, Domain verification, and Page usage agreements. The main content area is titled 'Credentials' and includes a 'Create credentials' button, a 'Delete' button, and a table of OAuth 2.0 client IDs. The table has columns for Name, Creation date, Type, and Client ID. One client ID is listed with a creation date of Aug 27, 2019 and a type of Web application.

<input type="checkbox"/>	Name	Creation date	Type	Client ID
<input type="checkbox"/>	[REDACTED]	Aug 27, 2019	Web application	[REDACTED]

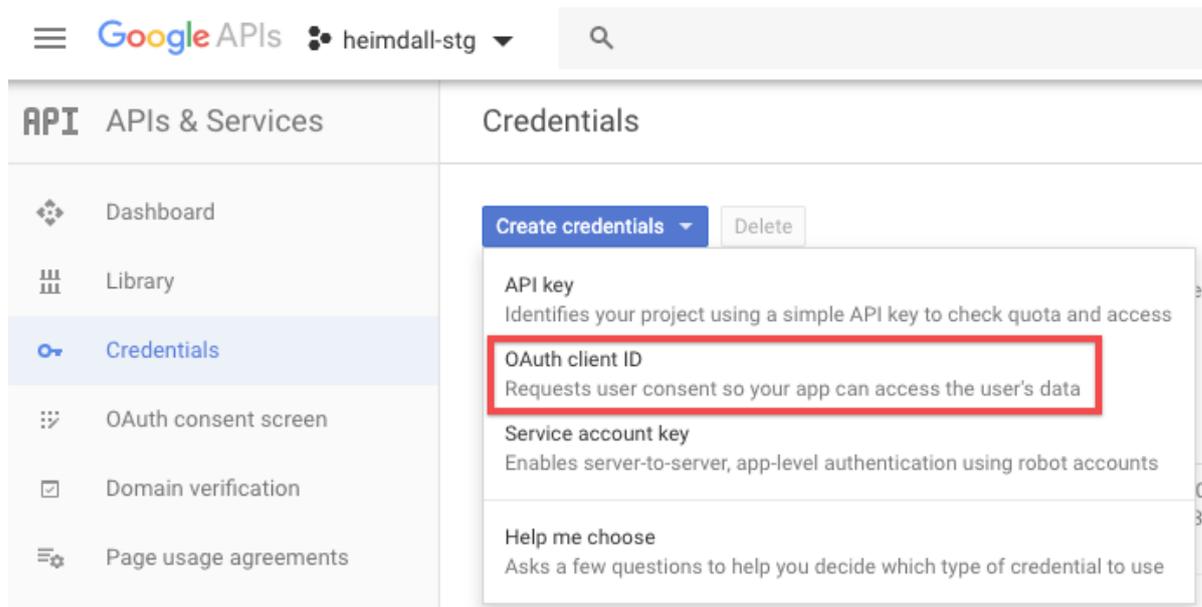
Select from   **NEW PROJECT** 

RECENT ALL

Name	ID
               	



2. Click Create Credentials, and then select OAuth client ID.



- If you are asked for a product name on the Consent screen, click Configure Consent Screen and follow the instructions in the below [Configuring a Consent Screen](#) section.
3. On the Create OAuth client ID page, click Web application, and enter a name for your application.
 4. In the Authorized JavaScript Origins field, add the following domains:

- o `https://staging.cloud-elements.com`
 - o `https://cloud-elements.com`
 - o `https://cloudelements.io`
 - o `https://staging.cloudelements.io`
5. In the Authorized Redirect URIs field, add `https://auth.cloudelements.io/oauth`

Your Create OAuth Client ID page should look like this:

Google APIs My Cool Application

Create OAuth client ID

For applications that use the OAuth 2.0 protocol to call Google APIs, you can use an OAuth 2.0 client ID to generate an access token. The token contains a unique identifier. See [Setting up OAuth 2.0](#) for more information.

Application type

- Web application
- Android [Learn more](#)
- Chrome App [Learn more](#)
- iOS [Learn more](#)
- Other

Name [?](#)

My Cool Application

Restrictions

Enter JavaScript origins, redirect URIs, or both [Learn More](#)

Origins and redirect domains must be added to the list of Authorized Domains in the [OAuth consent settings](#).

Authorized JavaScript origins

For use with requests from a browser. This is the origin URI of the client application. It can't contain a wildcard (`https://*.example.com`) or a path (`https://example.com/subdir`). If you're using a nonstandard port, you must include it in the origin URI.

<code>https://staging.cloud-elements.com</code>	
<code>https://cloud-elements.com</code>	
<code>https://cloudelements.io</code>	
<code>https://staging.cloudelements.io</code>	

`https://www.example.com`

Type in the domain and press Enter to add it

Authorized redirect URIs

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.

<code>https://auth.cloudelements.io/oauth</code>	
--	--

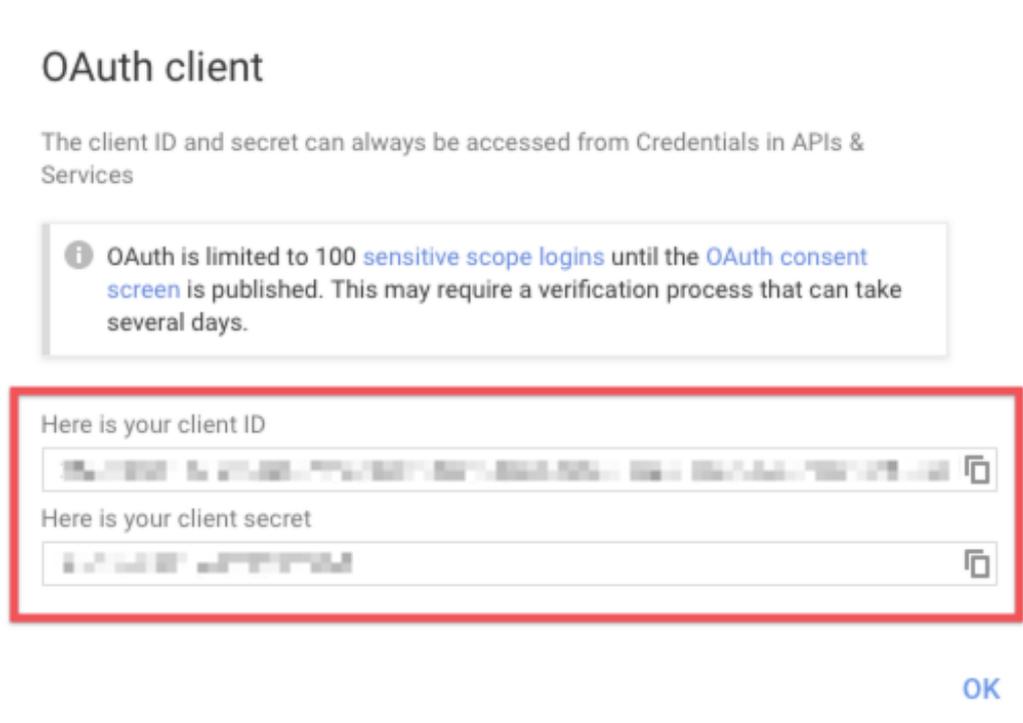
`https://www.example.com`

Type in the domain and press Enter to add it

Create Cancel

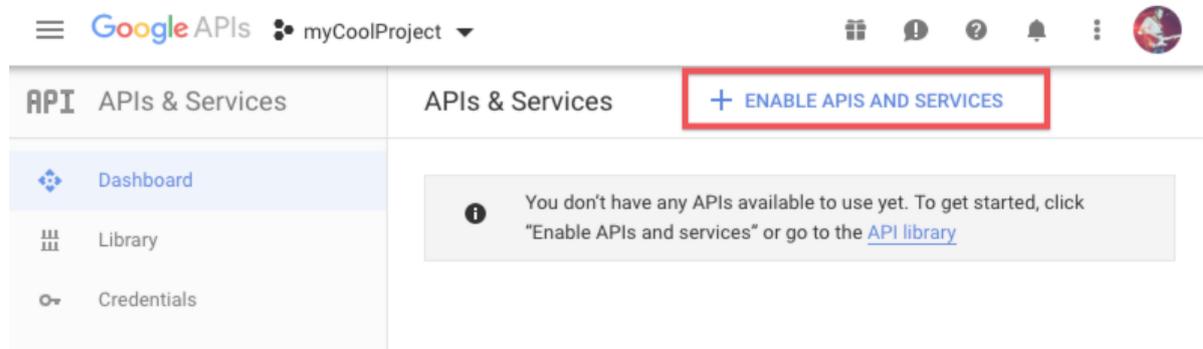
6. Click Create.

7. On the OAuth client window, record your client ID and secret, both of which you will use during authentication.

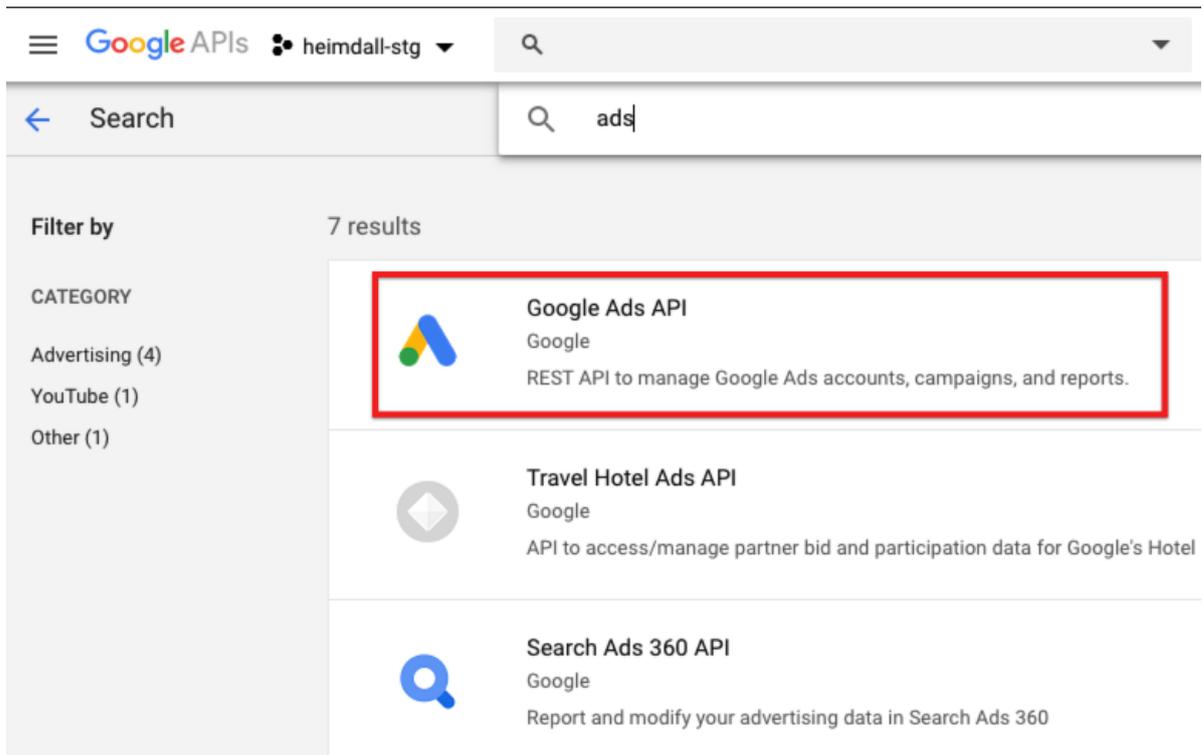


Enabling the API

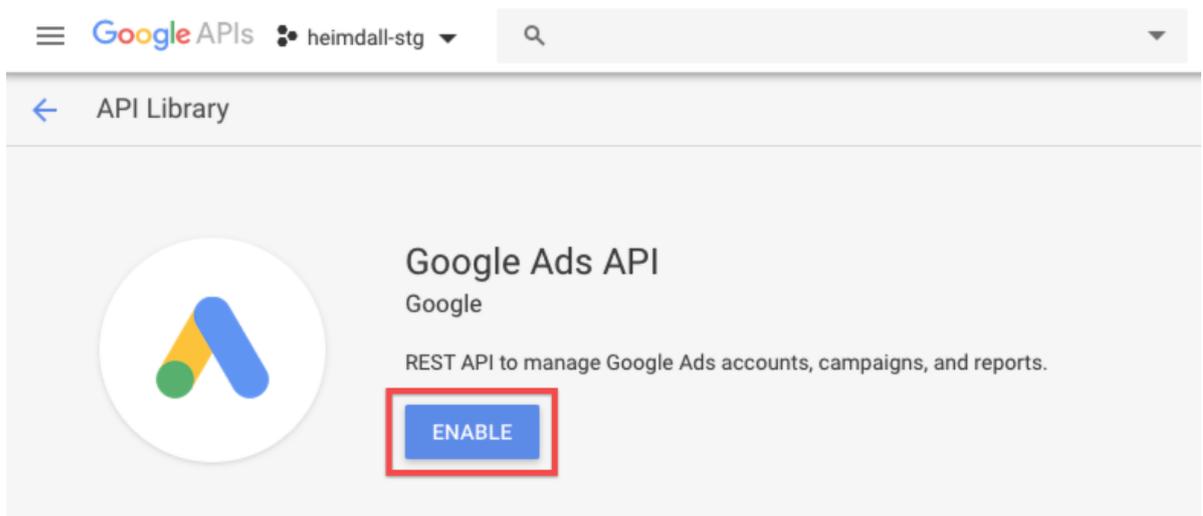
1. Navigate to the [Google API Console](#), and then click Enable APIs and Services.



2. On the API Library page, navigate to and select the Google Ads API.

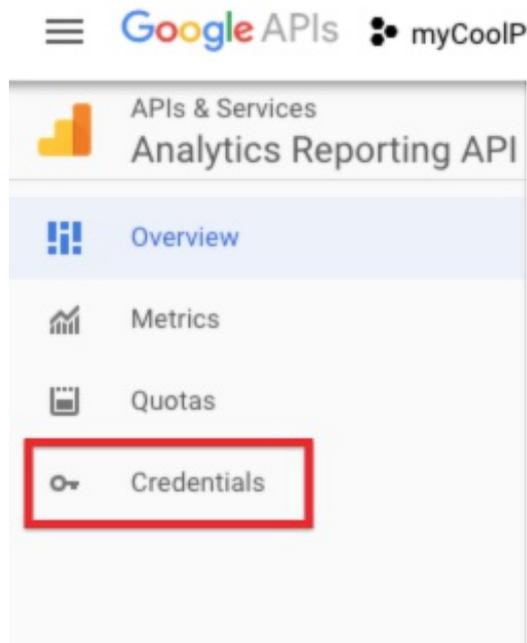


3. On the Google Ads API page, click Enable.

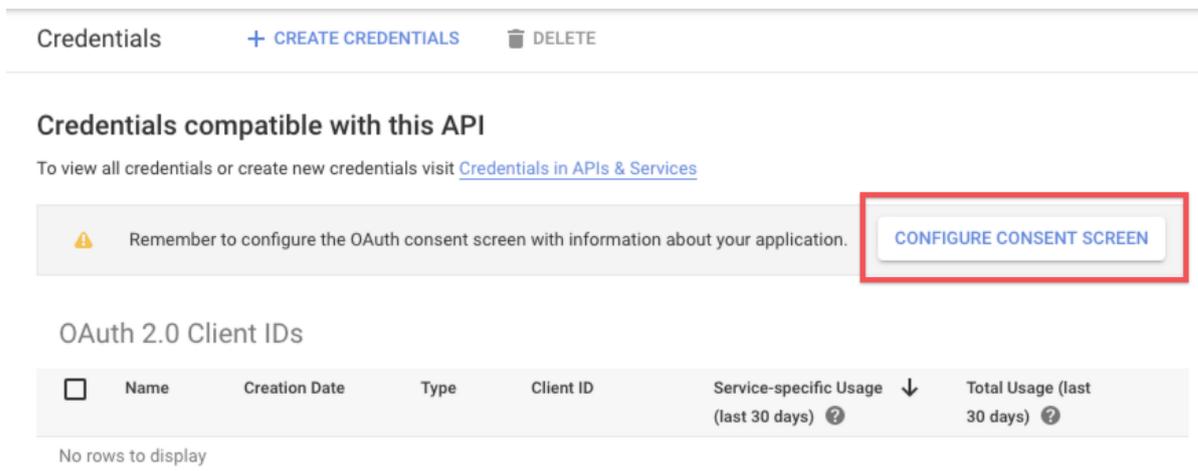


Configuring the Consent Screen

1. On the left-side navigation bar, click Credentials.



2. On the Credentials page, click Configure Consent Screen.



3. Enter a name for your application, and optionally add an application logo.
4. Under Scopes for Google APIs, click Add Scope and then click the option to manually paste your scopes. Enter the following scopes into the field, and then click Add:
 - o <https://www.googleapis.com/auth/analytics.manage.users.readonly>
 - o <https://www.googleapis.com/auth/analytics.manage.users>
 - o <https://www.googleapis.com/auth/analytics.edit>
 - o <https://www.googleapis.com/auth/analytics.readonly>
5. The Scopes for Google APIs section should look like this:

Scopes for Google APIs

Scopes allow your application to access your user's private data. [Learn more](#)

If you add a sensitive scope, such as scopes that give you full access to Gmail or Drive, Google will verify your consent screen before it's published.

 Because you've added a sensitive scope, your consent screen requires verification by Google before it's published. [Learn more](#)

email	
profile	
openid	
 ../auth/analytics.manage.users.readonly	
 ../auth/analytics.manage.users	
 ../auth/analytics.edit	
 ../auth/analytics.readonly	

Note: As explained in the above screenshot, your application will need to be verified by Google before it will be published. For more information about the application verification process, see [Google's documentation](#).

- In the Authorized Domains field, add `cloudelements.io` and `cloud-elements.com` as approved domains.

Authorized domains 

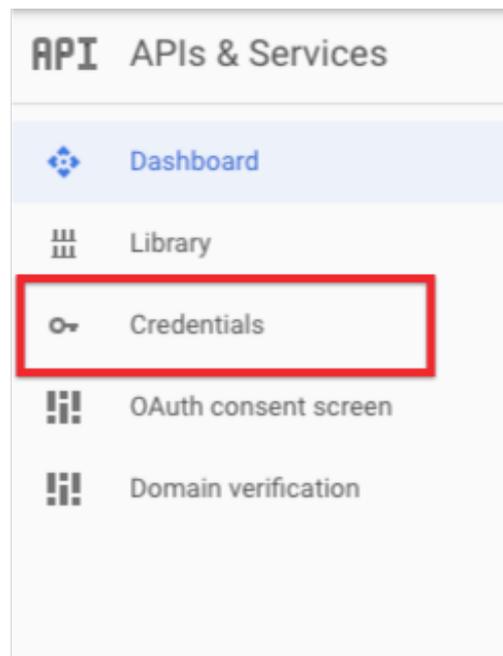
To protect you and your users, Google only allows applications that authenticate using OAuth to use Authorized Domains. Your applications' links must be hosted on Authorized Domains. [Learn more](#)

 Because you've added a sensitive scope, your consent screen requires verification by Google before it's published. [Learn more](#)

 cloudelements.io	
 cloud-elements.com	

Type in the domain and press Enter to add it

- Click Save, and then click Credentials from the left-side navigation bar.



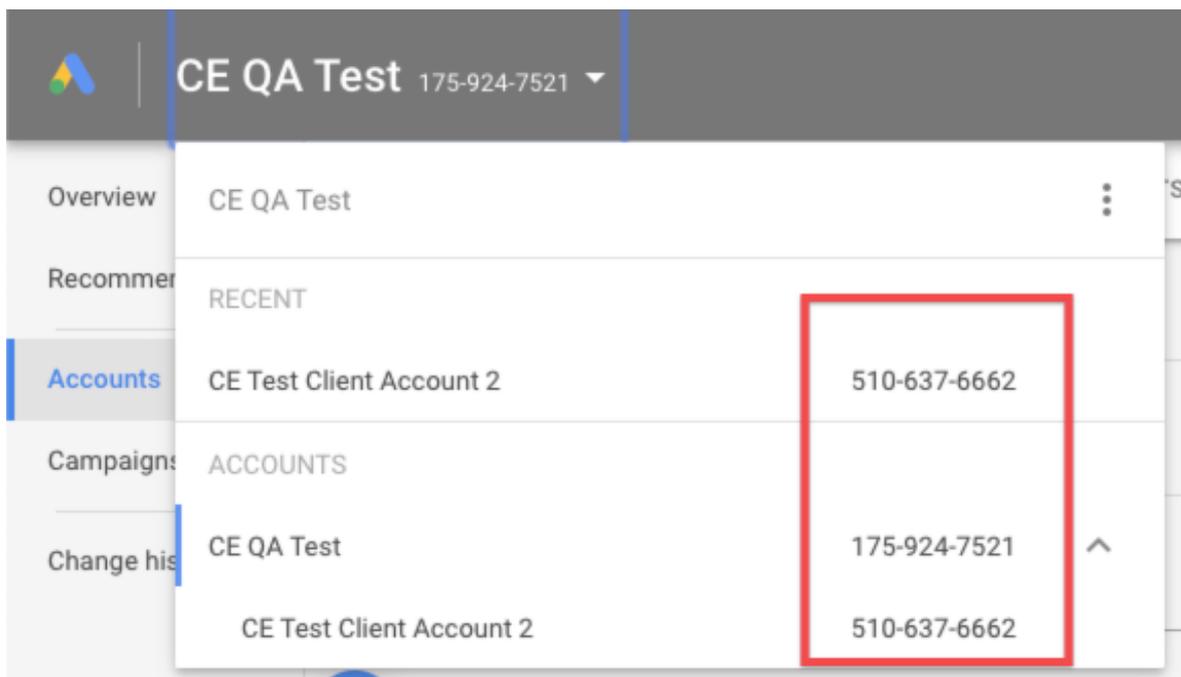
Identifying the Client Customer ID

1. In a web browser, navigate to <https://adwords.google.com/home> and sign in with your relevant Google credentials.

Note: If you have multiple Google Ads accounts, the **Use Google Ads as...** window appears, if so, select the relevant account from the list.

After selecting the relevant Google AdWords account, the AdWords interface appears.

2. In the AdWords interface, click the project selection dropdown and identify the account you want to connect to SAP Cloud Platform Open Connectors. The client customer ID value for each account is listed in the column to the right of the account names and is indicated in the example screenshot below.



Note the client customer ID, as you will need it in order to [authenticate a connector](#)

instance.
