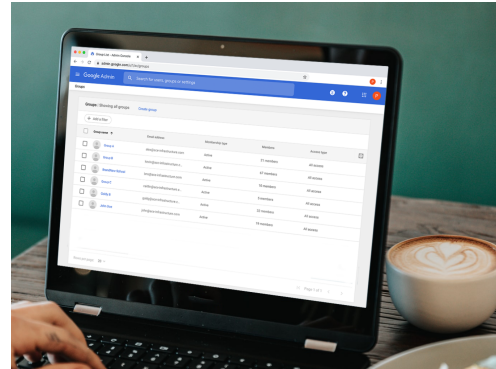




Getting started with the Chrome Browser Cloud Management **Postman API Integration**

Last updated Feb 2022



Decide on which Authorization method to use	2
Decide which APIs scopes (functionality) you want to use	2
Creating the API account in the Admin Console	3
Creating your project in console.cloud.google.com	4
Option 1: Authorize with Consent method	5
Option 2: Authorize with impersonation (service account)	8
Setting up Postman Integration	11
Verifying the connection to Chrome Browser Cloud Management	18

This document will guide you through the process of setting up the API integration with the [Postman API Platform](#). By using Postman you can accelerate the design, mocking, and testing of your scripts to make the process easier than ever to use [Chrome Browser Cloud Management](#) APIs. By using Cloud management APIs, you can push and pull data into the console at scale and enhance the level of reporting that you can gather from your enrolled browsers. This document assumes the following:

- Have access to the Google admin console
- A super admin account for the setup
- At least one (or more) device(s) enrolled into Chrome Browser Cloud Management for testing

Here are some useful links:

[Chrome Browser Enterprise Github](#)

[Readme file for the API](#)

[Google Cloud Platform Console](#)

[Postman API Platform](#)

Decide on which Authorization method to use

Google provides two different methods to authorize the connection between Chrome Browser Cloud Management, the API and Postman. Depending on how you are going to be using the API would point to which method to use.

[Authorize with consent](#)

Each time you request a token, the admin will be prompted to authorize the request via the API.

- Postman scripts are developed using this authorization method. For validating API calls, this is the recommended method for your integration development work

[Authorize with impersonation or a service account](#)

Authorize with the service account once and then will not be prompted when the request is made

- Supports server-to-server interactions such as those between a web application and a Google service. Ideal for production server-to-server interactions.

Decide which APIs scopes (functionality) you want to use

The API provides four different APIs that you can enable. Depending on what requests or actions you want to take via API could determine which scopes you enable. You can also enable them with read or write access (or both).

Complete descriptions of what each can do is located in the [Github documentation](#).

Here is a brief overview of each one:

Scope	Description
Write scopes	
https://www.googleapis.com/auth/admin.directory.device.chromebrowsers	Chrome Browser Cloud Management Lets you view and modify enrolled browsers and enrollment tokens
https://www.googleapis.com/auth/admin.directory.orgunit	Org Units - lets you view and modify organizational units

https://www.googleapis.com/auth/chrome.management.policy	Chrome Policy - lets you view and modify Chrome policies for devices and users
Read only scopes	
https://www.googleapis.com/auth/admin.directory.device.chrome.browsers.readonly	Chrome Browser Cloud Management Get detailed information on enrolled browsers and enrollment tokens (read-only)
https://www.googleapis.com/auth/chrome.management.reports.readonly	Reports - Chrome versions and installed apps (read-only)
https://www.googleapis.com/auth/chrome.management.appdetails.readonly	App Details - get detailed information about requested or specified apps (read-only)
https://www.googleapis.com/auth/chrome.management.policy.readonly	Chrome Policy - lets you view Chrome policies for devices and users (read-only)
https://www.googleapis.com/auth/admin.directory.orgunit.readonly	Org Units - lets you view organizational units (read-only)
https://www.googleapis.com/auth/admin.reports.audit.readonly	Admin Console Reports - lets you view activities done by administrators using the Admin console and OAuth token activities (read-only)

Creating the API account in the Admin Console

In order to make API calls into the admin console, an API user account needs to be created. This account requires all of the specific API privileges in order to push and pull data from the Admin console.

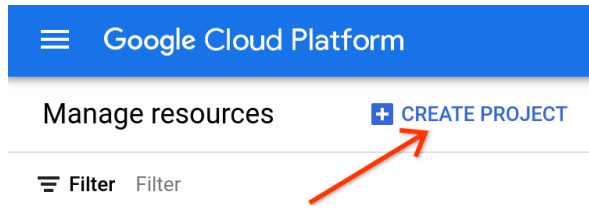
1. Go to the admin console and sign in with an admin account with the necessary privileges to enable API access and create new user accounts (usually this is a super admin account).
2. Go to Admin Roles -> Create Role
3. Create a name for your API role and select desired privileges under "Organizational Units", "Chrome Management" and then hit the Create Role button.
4. Create a user account that you will assign the API role to under Directory>Users>add new user
5. Give the API account a name and create an email address and click the add new user button.
6. Go to the user account that you created in the previous step and go to -> Admin roles and Privileges -> and select the role that you created in Step 3.
 - a. Please note that propagation of this new permission may take a few minutes

Creating your project in console.cloud.google.com

1. Open console.cloud.google.com

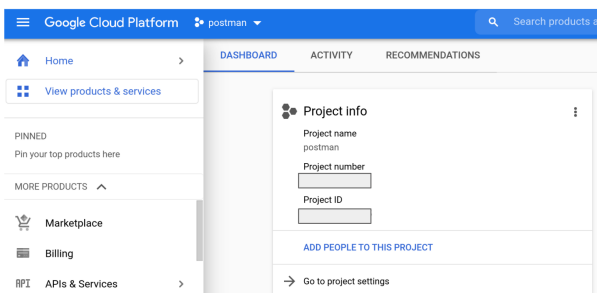
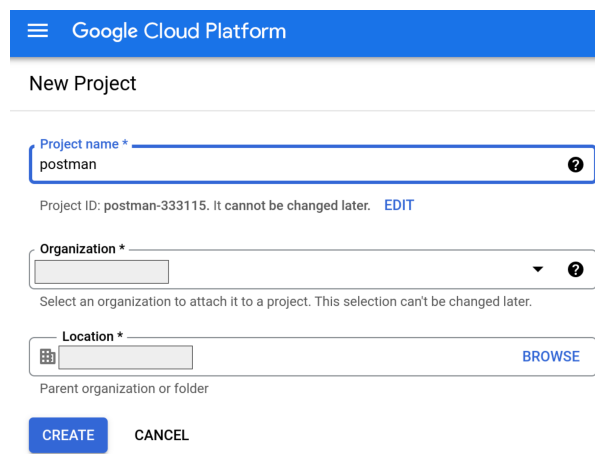
Make sure that you are signed into the cloud console with the new API role account that you created in the previous section.

2. Press the create project button.



3. Enter in the Project name

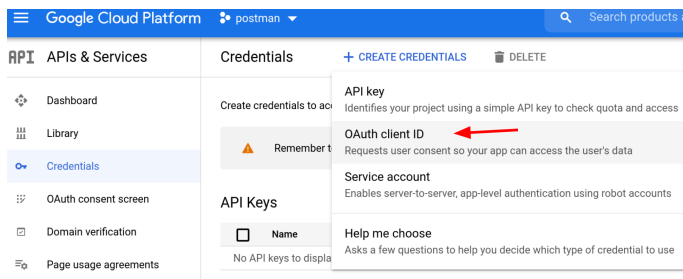
(can be whatever you choose) and the location should match the domain of your admin console. Hit create.



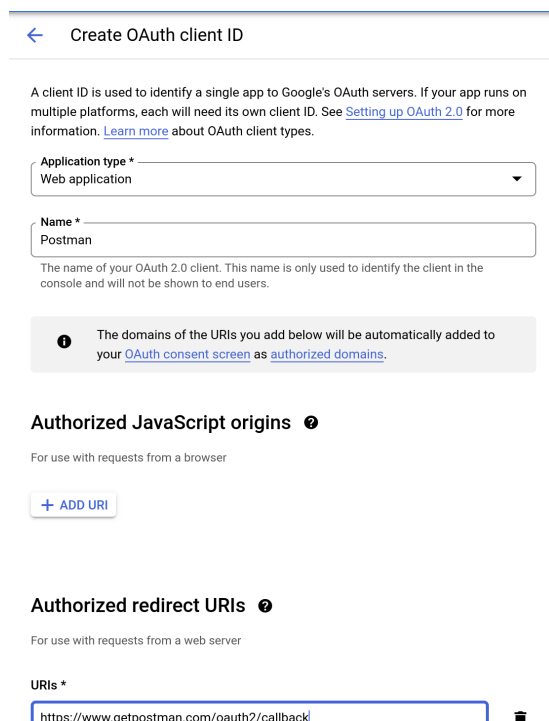
4. Once created make sure that you have that project selected.

Option 1: Authorize with Consent method

1. Open console.cloud.google.com, select the project that you created in the previous steps and browse to APIs and Services> Credentials>and hit the Create Credentials button and select OAuth client ID



2. Select web application for Application type web application and provide a name.
3. For authorized redirect URI enter in the following:
 - a. <https://www.getpostman.com/oauth2/callback>
 - b. <https://oauth.pstmn.io/v1/browser-callback>
4. Hit the create button.

A screenshot of the 'Create OAuth client ID' form in the Google Cloud Platform console. The form is titled 'Create OAuth client ID' and has a back arrow on the left. Below the title, there is a paragraph explaining that a client ID is used to identify a single app to Google's OAuth servers. The form contains several fields: 'Application type' (a dropdown menu set to 'Web application'), 'Name' (a text input field containing 'Postman'), and a note stating that the domains of the URIs added below will be automatically added to the OAuth consent screen as authorized domains. There are two sections for adding URIs: 'Authorized JavaScript origins' (for use with requests from a browser) and 'Authorized redirect URIs' (for use with requests from a web server). The 'Authorized redirect URIs' section has a text input field containing the URL 'https://www.getpostman.com/oauth2/callback' and a trash icon to its right.

5. On the OAuth client created window, click the download JSON button.
6. Click on the OAuth consent screenConfigure consent screen
When you use OAuth 2.0 for authorization, your app requests authorizations for one or more scopes of access from a Google Account. Google displays a consent screen to the user including a summary of your project and its policies and the requested scopes of access.
There are two types:
 - a. **Internal** is just for users within your Google Workspace users. Requires Google Workspace or Google Identity to function.
 - b. **External** is available to any test user with a Google Account. Your app will start in testing mode and will only be available to users you add to the list of test users. Once your app is ready to push to production, you may need to verify your app.
7. Enter in the App name, the user support email (this is your admin account email in the admin console) and you can also choose to display a custom logo.
8. Under Authorized domains enter in **getpostman.com**
9. Enter in an email address that you want Google to notify you about changes in your project and hit save and continue.

App information

This shows in the consent screen, and helps end users know who you are and contact you

App name *
Postman API

The name of the app asking for consent

User support email *

For users to contact you with questions about their consent

App logo BROWSE

Upload an image, not larger than 1MB on the consent screen that will help users recognize your app. Allowed image formats are JPG, PNG, and BMP. Logos should be square and 120px by 120px for the best results.

Authorized domains ?

When a domain is used on the consent screen or in an OAuth client's configuration, it must be pre-registered here. If your app needs to go through verification, please go to the [Google Search Console](#) to check if your domains are authorized. [Learn more](#) about the authorized domain limit.

getpostman.com

[+ ADD DOMAIN](#)

Developer contact information

Email addresses *

These email addresses are for Google to notify you about any changes to your project.

10. Click the **Add or Remove Scopes** button to add the specific scopes manually. They are all listed in this [list of scopes in the Chrome Browser Enterprise Github](#). Hit save and continue.

Note that the list contains both read only and full access. Pick and choose which scopes you need for your specific use case.

× Update selected scopes

i Only scopes for enabled APIs are listed below. To add a missing scope to this screen, find and enable the API in the [Google API Library](#) or use the Pasted Scopes text box below. Refresh the page to see any new APIs you enable from the Library.

Filter Enter property name or value **?**

<input type="checkbox"/>	API ↑	Scope	User-facing description
<input checked="" type="checkbox"/>		.../auth/admin.directory.device.chromebrowsers	See and manage Chrome browsers under your organization
<input checked="" type="checkbox"/>		.../auth/admin.directory.device.chromebrowsers.readonly	See Chrome browsers under your organization
<input checked="" type="checkbox"/>		.../auth/admin.directory.orgunit	View and manage organization units on your domain
<input checked="" type="checkbox"/>		.../auth/admin.directory.orgunit.readonly	View organization units on your domain
<input checked="" type="checkbox"/>		.../auth/admin.reports.audit.readonly	View audit reports for your G Suite domain
<input checked="" type="checkbox"/>		.../auth/chrome.management.appdetails.readonly	See detailed information about apps installed on Chrome browsers and devices managed by your organization
<input checked="" type="checkbox"/>		.../auth/chrome.management.reports.readonly	See reports about devices and Chrome browsers managed within your organization
<input checked="" type="checkbox"/>		.../auth/chrome.management.policy.readonly	See policies applied to Chrome OS and Chrome Browsers managed within your organization
<input checked="" type="checkbox"/>		.../auth/chrome.management.policy	See, edit, create or delete policies applied to Chrome OS and Chrome Browsers managed within your organization

11. **Optional** - Add a user(s) that can access while the publishing status is listed as “Testing”. Enter in the account that you used to create the project and any others that you want to provide access to by hitting the add users button.

12. Hit **save and continue**, review the summary screen and hit the back to dashboard button.

13. If you choose at this time you can **publish your App**.

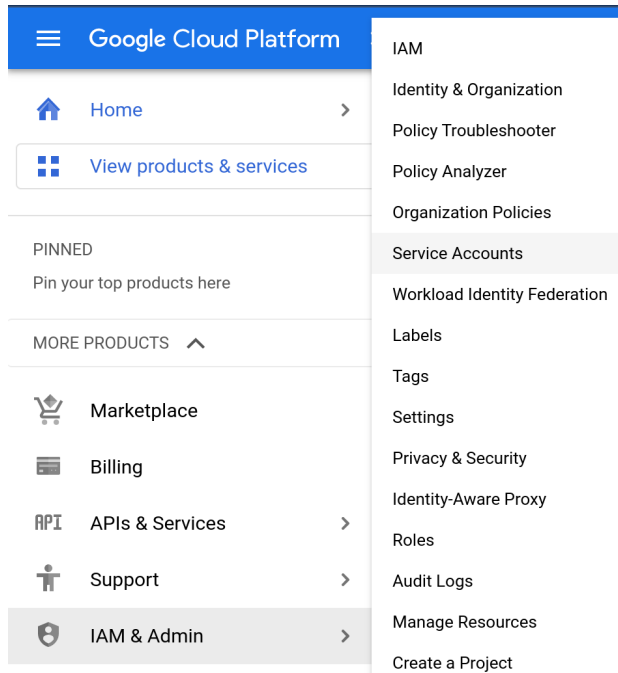
- a. Once you set your app status as **"In production,"** your app will be available to anyone with a Google Account. Depending on how you configure your OAuth screen, you may have to submit your app for verification



Option 2: Authorize with impersonation (service account)

Creating a service account (authorize with impersonation)

1. Open the [Service accounts page](#) under IAM & Admin. If prompted, select a project.



2. Click + Create Service Account, enter a name and description for the service account



3. You can use the default service account ID, or choose a different, unique one.
When done click Create.

1 Service account details

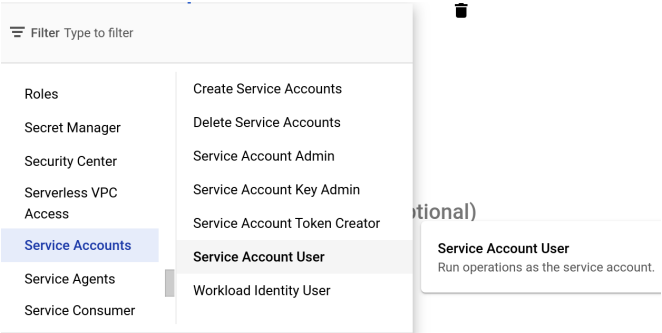
Service account name *
postman
Display name for this service account

Service account ID *
postman @ [] .com X ↻

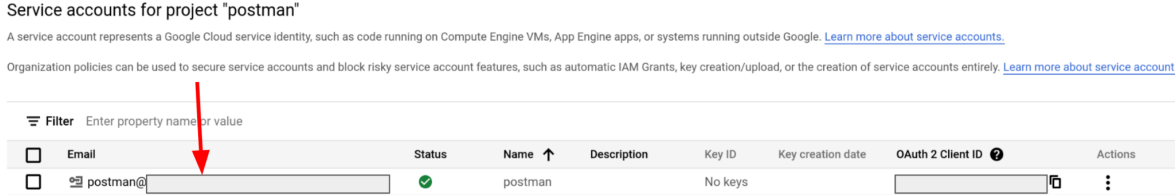
Service account description
Describe what this service account will do

CREATE AND CONTINUE

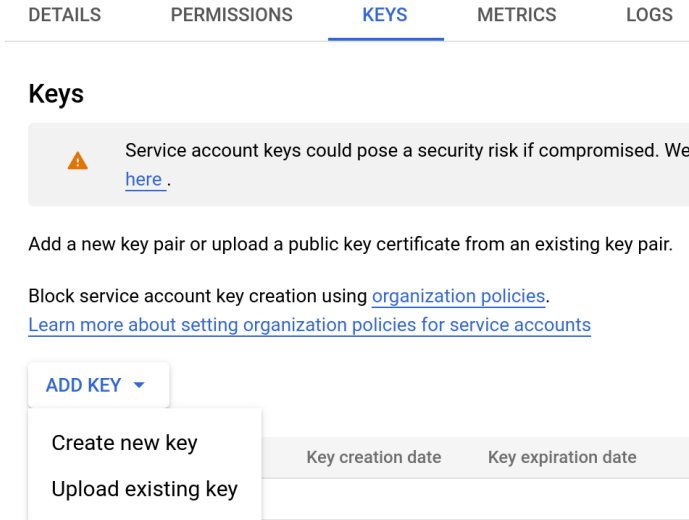
- 4. In the Service account permissions make sure to grant the service account the "Service Account User" role. Hit continue.
- 5. On the Grant users access to this service account screen, optionally add users that you want to grant access and hit the done button.



- 6. Select the service account that you just created in the next screen



- 7. Select the Keys tab at the top and select Create new key from the dropdown menu and select JSON as the key type.



- Your new public/private key pair is generated and downloaded to your machine; it serves as the only copy of this key. It provides access to your cloud console, so **store it securely**.
- Click on the Details tab** of your selected service account and copy the email address which is in the form <project-name>-<id>@<project-name>.iam.gserviceaccount.com.

DETAILS PERMISSIONS KEYS METRICS LOGS

Service account details

Name
postman SAVE

Description SAVE

Email
postman@


- Navigate to the APIs & Services page under the project you created earlier in GCP. Click on the + Enable APIs and Services

Google Cloud Platform postman ▼

API APIs & Services APIs & Services + ENABLE APIS AND SERVICES

Enabled APIs & services

- Do a search for the following APIs, select them one by one and hit the Manage button

 **Chrome Management API**
Google

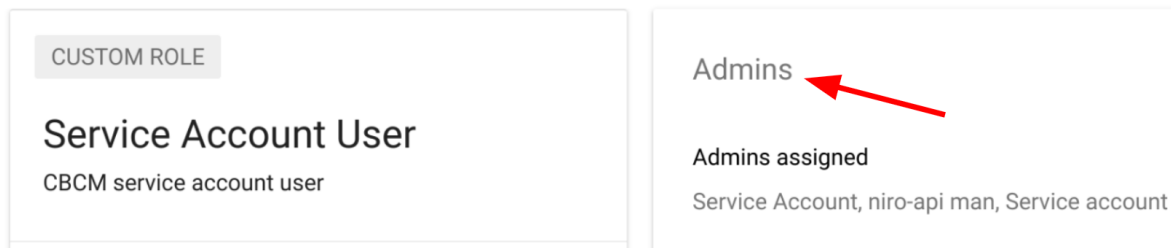
The Chrome Management API is a suite of services that allows Chrome administrators to view, manage...

[MANAGE](#) [TRY THIS API ↗](#)

- Chrome Management API =
<https://console.developers.google.com/apis/api/chromemanagement.googleapis.com/overview>
- Chrome Policy API =
<https://console.developers.google.com/apis/api/chromepolicy.googleapis.com/overview>
- Admin SDK API =
<https://console.developers.google.com/apis/api/admin.googleapis.com/overview>

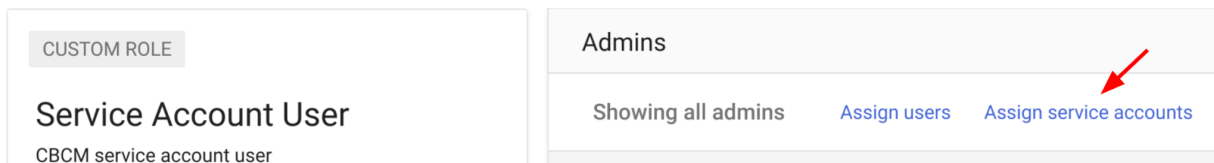
12. Log back into the [Google admin console](#) with your super admin account.
13. Browse to [Account>Admin roles>](#) and select the service account role that you created in [Creating your API account section](#).
14. Click on the Admins card at the top of the page.

Admin roles > Service Account User



The screenshot shows the Google Admin console interface. On the left, there is a card for the 'Service Account User' role, labeled as a 'CUSTOM ROLE' and 'CBCM service account user'. On the right, there is an 'Admins' card with a red arrow pointing to it. Below the 'Admins' card, it says 'Admins assigned' and lists 'Service Account, niro-api man, Service account'.

15. Click the assign service account.



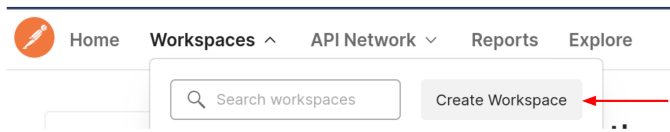
The screenshot shows the Google Admin console interface. On the left, there is a card for the 'Service Account User' role, labeled as a 'CUSTOM ROLE' and 'CBCM service account user'. On the right, there is an 'Admins' card with a red arrow pointing to the 'Assign service accounts' button. Below the 'Admins' card, it says 'Showing all admins' and lists 'Assign users' and 'Assign service accounts'.

16. Enter in the email address that you copied from step 9 and click the add button.

Setting up Postman integration

Setting up an account

1. Go to [Postman.com](#) and set up a new account or use an existing one.
2. Once your account is setup, go to **Workspaces>Create Workspace** and give it a name and select the level of visibility you want and hit the Create Workspace button



Create workspace

Name

Summary

Add a brief summary about this workspace.

Visibility

Determines who can access this workspace.

Personal

Only you can access

Private

Only invited team members can access

Team

All team members can access

Public

Everyone can view

Create Workspace

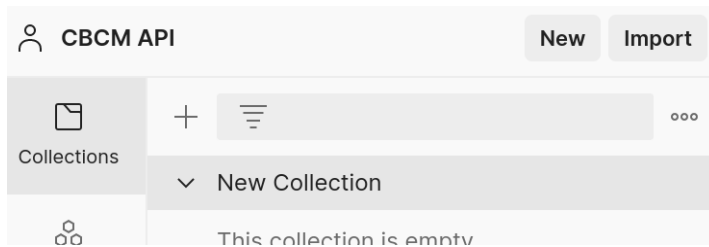
Cancel

Importing a collection from the Chrome Enterprise Github

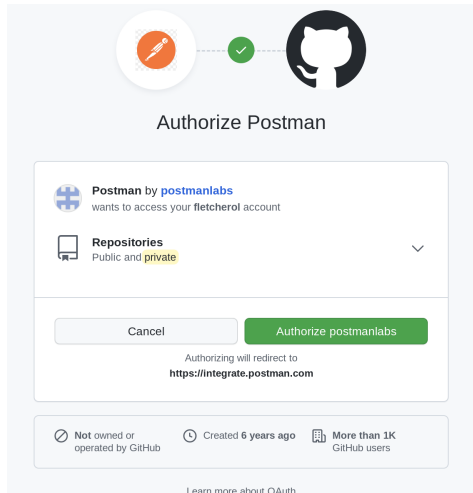
It is recommended to import the collections that are provided in the [Chrome Enterprise Github](#). The provided collections provide sample scripting for common use cases for calls to Chrome Browser Cloud Management. You can import them via the file method or link to your Github repository.

Importing via Code repository

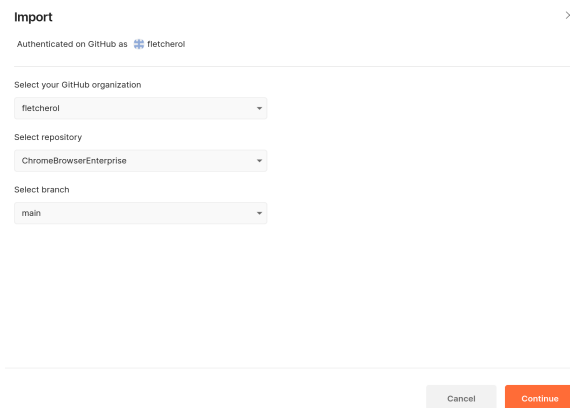
1. Log into your Github account.
2. Go to <https://github.com/google/Chrome BrowserEnterprise> and hit the fork button to add it to your repository.
3. In Postman with your Workspace selected, click on the import button.



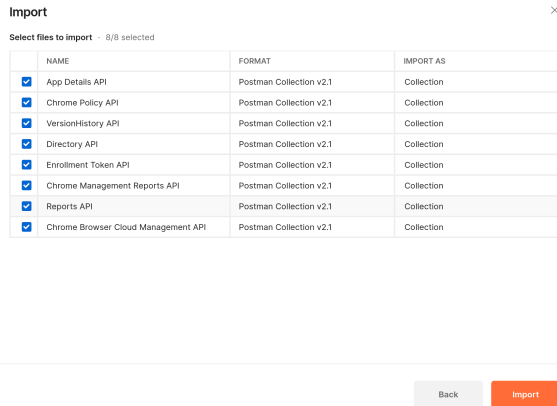
4. Click on Code repository> Github
5. Log into your Github account and click on the authorize postmanlabs button and confirm your password.



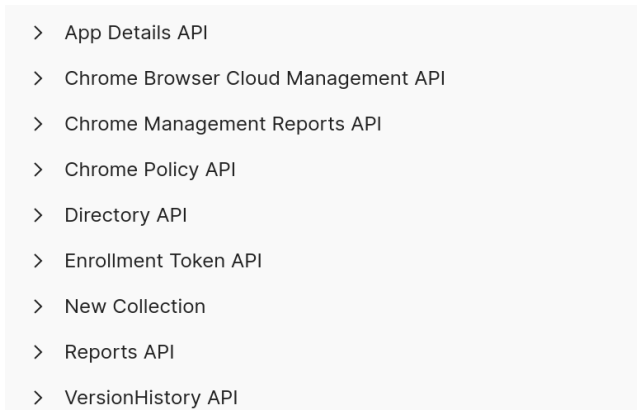
6. Once this is complete in the import section you should see the ChromeBrowserEnterprise selectable in the dropdown menu under Select repository. Hit the continue button.



7. Select which collections you want to import and hit the import button.

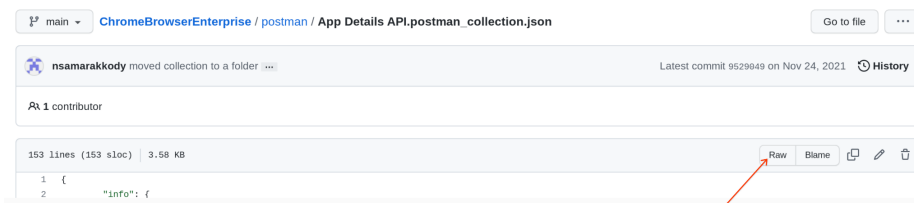


8. Once complete you should see all of the collections included in your Postman workspace



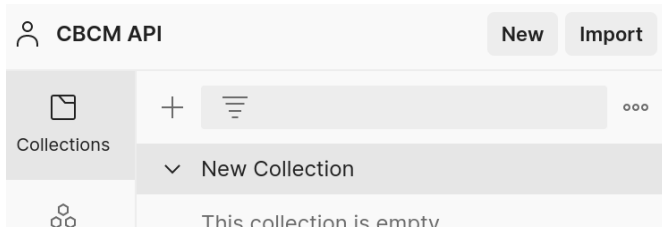
Importing via Upload files

1. Browse to the [Chrome Enterprise Github](#) and select the collection that you want to import.
2. Click on the raw button



3. Right click and save as .json.

- In Postman with your Workspace selected, click on the import button.



- Hit the Upload Files button and select the JSON that you downloaded in the previous step and hit the import button



The following steps are only needed for the Authorize with Consent method only:

- Once the Collection shows up, you will need to fill in the sections highlighted in red.

Auth URL ⓘ	<input type="text" value="{{auth_uri}}"/>
Access Token URL ⓘ	<input type="text" value="{{token_uri}}"/>
Client ID ⓘ	<input type="text" value="{{client_id}}"/>
Client Secret ⓘ	<input type="text" value="{{client_secret}}"/>

- For Auth URL enter in the following:
<https://accounts.google.com/o/oauth2/auth>
- For Access Token URL enter in the following:
<https://oauth2.googleapis.com/token>

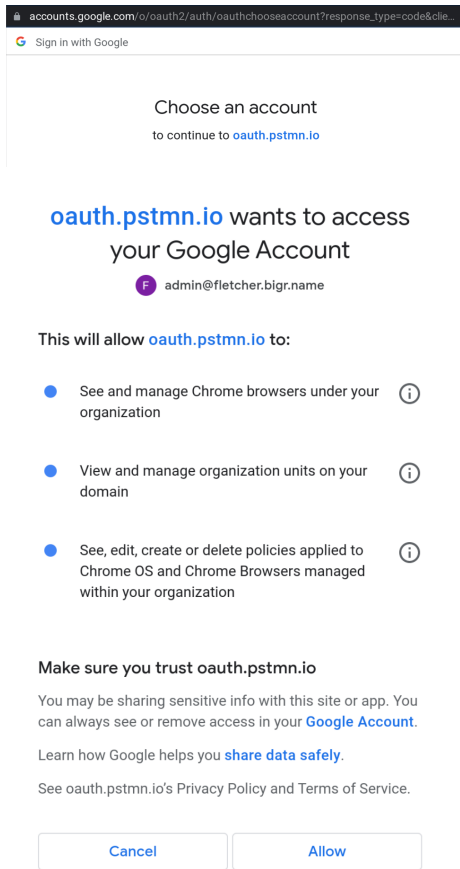
9. Get the client id and client secret from the API and services section of <https://console.cloud.google.com/> (direct link) and enter them into Postman.

Client ID for Web application DOWNLOAD JSON RESET SECRET DELETE

Name *
Postman
The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

Client ID	Client secret	Creation date

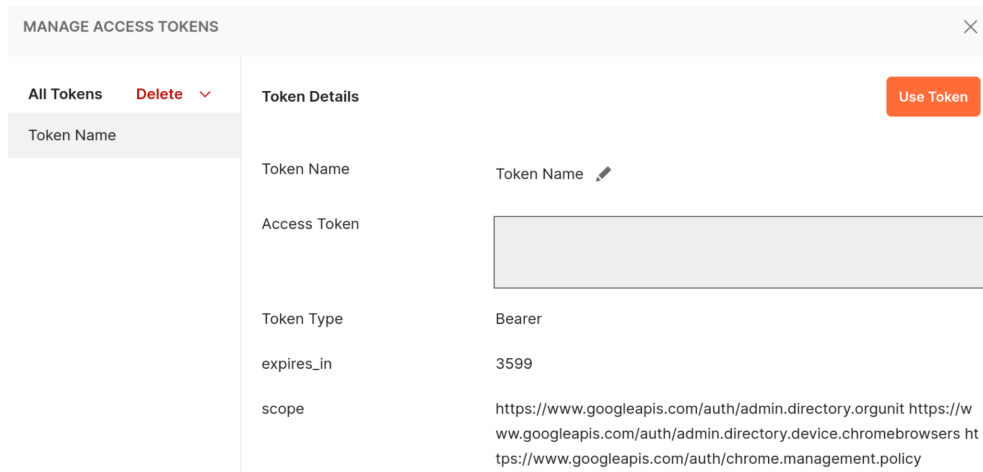
- a. Note that it is recommended to add in these values as variables to keep sensitive data secure and make it easier to add the data to multiple collections as you import them.
- b. If you want to use the same variables in multiple collections you need to declare them as global variables.
- More information about Postman [variables is located here](#).
10. Get the client id and client secret from the API and services section of <https://console.cloud.google.com/> (direct link) and enter them into Postman.
11. If you are using the browser version of Postman, the callback URL this is prefilled in, on the desktop version you might need to input to put the following:
<https://www.getpostman.com/oauth2/callback>
12. Hit the Get New Access Token button.
13. When you hit the get new access token button, you will be prompted to authenticate using your admin credentials for Chrome Browser Cloud Management.
Make sure that this account has API privileges granted for the scopes that you have added.



14. Once completed you will be presented with your new token that you will be using to authenticate requests to the APIs.

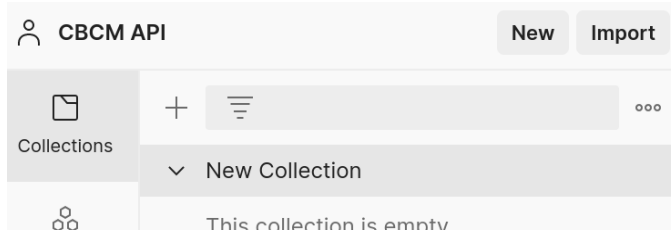
Hit the use token button and the token information will be prefilled into the collection for you.

Tokens are valid for one hour by default.



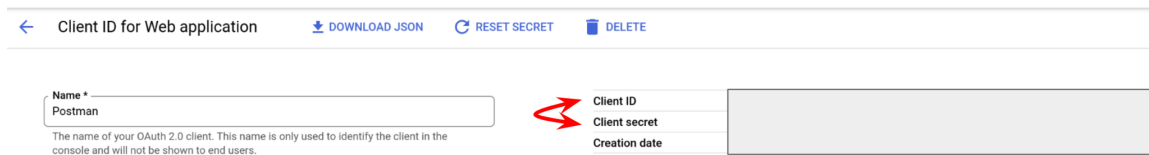
Creating a new collection

1. Under the Collections tab hit the plus mark to create a new collection



The following steps are only needed for the Authorize with Consent method only:

2. Select OAuth2 for the type of Authorization
3. For the callback URL enter in the following:
 - a. If using the browser version of Postman this is prefilled in, on the desktop version you need to put the following: <https://www.getpostman.com/oauth2/callback>
4. For Auth URL enter in the following: <https://accounts.google.com/o/oauth2/auth>
5. For Access Token URL enter in the following: <https://oauth2.googleapis.com/token>
6. Get the client id and client secret from the API and services section of <https://console.cloud.google.com/> ([direct link](#)) and enter them into Postman.



Note that it is recommended to add in these values as variables to keep sensitive data secure and make it easier to add the data to multiple collections as you import them.

More information about Postman [variables is located here](#).

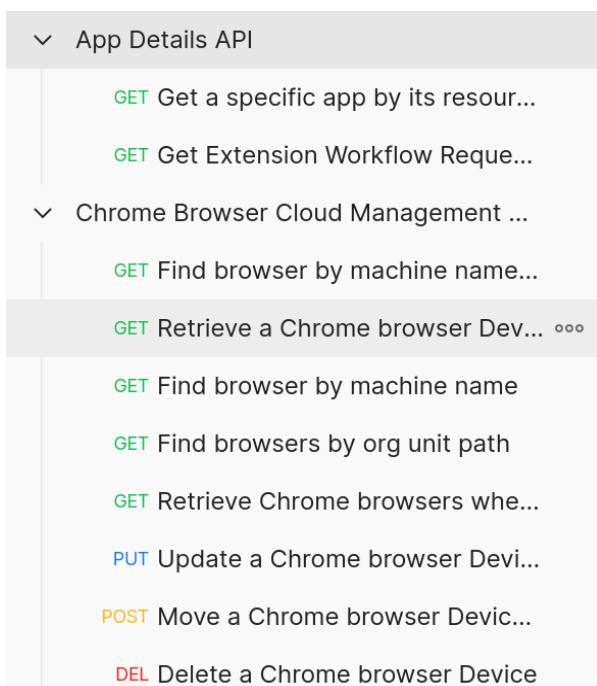
7. Add in the Cloud management API specific scope(s) that you want ([reference this section for more information](#)), each scope needs to be separated by a space.
 8. Hit the Get New Access Token button.
-

Verifying the connection to Chrome Browser Cloud Management

Each collection that is present in the Chrome browser Enterprise Github have some example scripts to get you started, with examples for GET, PUT, POST and DEL requests. Once you have imported the collection, you can run one of these to make sure that the Postman is able to make requests to Cloud management.

1. **Click on the collection that you imported.**

The following steps (2-6) are only needed for the Authorize with Consent method only:



2. **Confirm** that the Auth URL, Access Token URL, Client ID and Client Secret and scope sections are filled in either with the values or with a variable.

3. Click on the Get New Access Token button

App Details API

Authorization ● Pre-request Script Tests Variables ●

This authorization method will be used for every request in this collection. You can override this by specifying one in the request

ya29.A0ARrdaM_ydeit346V0Z ...

Header Prefix ⓘ Bearer

Configure New Token

Configuration Options ● Advanced Options

Token Name AppDetailsApiToken

Grant Type Authorization Code

Callback URL ⓘ https://oauth.pstmn.io/v1/browser-call

Auth URL ⓘ {{Auth URL}}

Access Token URL ⓘ {{Access Token URL}}

Client ID ⓘ {{Client ID}}

Client Secret ⓘ {{Client Secret}}

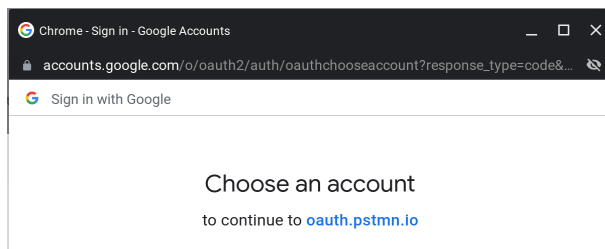
Scope ⓘ https://www.googleapis.com/auth/cl...

State ⓘ State

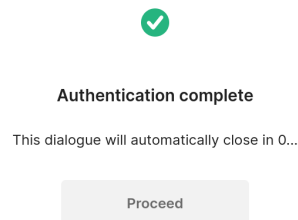
Client Authentication Send as Basic Auth header

Get New Access Token

4. Choose the account that you want to auth to



5. You will receive the authorization complete dialog if all is correct



6. Click on the Use Token button.

MANAGE ACCESS TOKENS

All Tokens Delete

Token Name

AppDetailsApiToken

AppDetailsApiToken

chromebrowsers

AppDetailsApiToken

Token Details

Use Token

Token Name AppDetailsApiToken

Access Token

Token Type Bearer

expires_in 3599

scope https://www.googleapis.com/auth/chrome.management.appdetails.readonly

7. Select the request that you want to make and click on the send button

App Details API / Get a specific app by its resource name

GET https://chromemanagement.googleapis.com/v1/customers/my_customer/...

Send

8. The response will show down below and you can change from JSON to other formats if you wish via the dropdown menu.

Body Cookies Headers (13) Test Results Status: 200 OK Time: 934 ms Size: 4.54 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {"name": "customers/C033fmrfg/apps/chrome/nckgahadagoaajjgafhacjanaoiihapd",
2  "displayName": "Google Hangouts",
3  "description": "Use Hangouts to keep in touch. Message friends, start free video or voice calls, and hop on a conversation with one person or a group.\n * Include all your
4  friends with group chats for up to 150 people.\n * Say more with photos, videos, maps, emoji, stickers, and animated GIFs.Turn any conversation into a free group video call
with up to 10 friends.\n * Keep in touch with friends across Android, iOS, and the web, and sync chats across all your devices.\n * Message friends anytime, even if they're
offline.\n * Snooze your notifications so you can respond later.\n * See what you chatted about in the past, including shared photos and your video call history.\n * Keep a
record of any conversation for just a short period of time by turning history off.\n * Connect your Google Voice account to make calls, send and receive SMS, and access
your voicemail.\n\nHangouts Chrome extension:\n * Use Hangouts and get notifications as you move from tab to tab in Chrome, or even without a Chrome window open.\n *
Position Hangouts anywhere on your screen, even if you have more than one monitor. Keep conversations in a single window or pop out the important ones. \n * View and
continue your conversations across devices.\n * Get notifications just once. After you see an alert, it'll be removed on other devices.\n\nNotes:\n * Unlike the Chat for
Google app, Hangouts doesn't support "invisible status".\n * Mobile carrier and ISP charges may apply.",
5  "appId": "nckgahadagoaajjgafhacjanaoiihapd",
6  "revisionId": "2020.803.419.1",
7  "tvpe": "CHROME",
```