

# Setup With Google for Admins

## Overview

MyWhistleBox by default will send out requests emails (Upload, Signature etc.) using it's own SMTP mail server. This has the side effect of emails "coming from" mywhistlebox.com and not the customers email account. Using the MyWhistleBox email settings, the default can be overridden to use an email server of the customers choice. Using the Google connector, you can connect to Google Mail via OAuth2.

This article will cover the required steps necessary for an administrator to configure the connector for outbound email messages.

## Prerequisites

- You must have a MyWhistleBox Professional level plan.
- You must have an active Google Workspace or Google Apps account to configure the Google connector for your instance.

## Steps to Complete

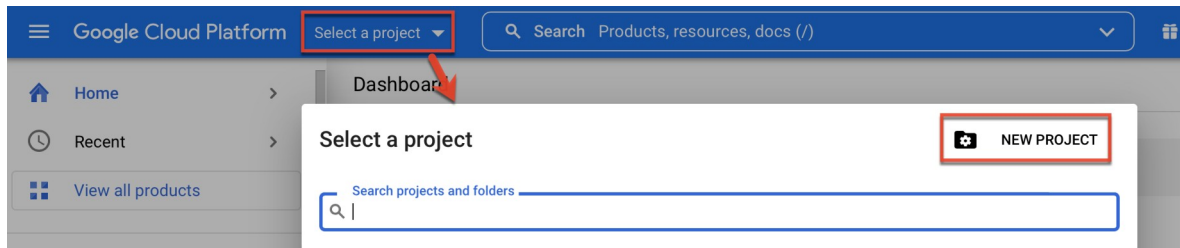
Before users may connect their Google accounts to MyWhistleBox, an admin user must complete the initial, system-wide Google connector configuration. The following sections explain how the administrator can acquire the proper API credentials from Google and use those credentials to set up the connector in MyWhistleBox.

## Enabling Google APIs and Creating Credentials

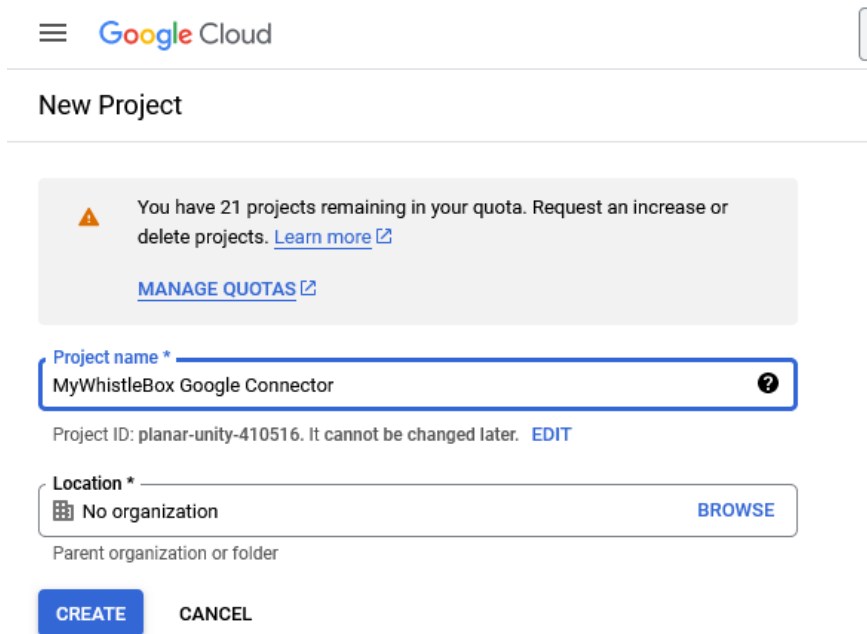
To set up the MyWhistleBox Google email connector, you must first enable the appropriate Google APIs and create credentials to obtain a Client ID and Client Secret.

1. Using your web browser, navigate to the Google Developers site (<https://console.cloud.google.com>).
2. Sign in using the Google account under which you would like to register the application.
3. Create a new project:

Click "Select a project" in the navigation bar. Click "New Project" in the Select a project window.



Enter a project name of your choice (e.g., MyWhistleBox Google Connector) and then click "Create".



4. Click the Google Cloud Platform logo in the top left of the screen and ensure that you are working in your newly created project (e.g., MyWhistleBox Google Connector). Click “**Api’s and Services**” from the Quick Access buttons.

[+ Create a VM](#) [+ Run a query in BigQuery](#) [+ Create a GKE cluster](#) [+ Create a storage bucket](#)

Quick access

API APIs & Services IAM & Admin Billing  
Cloud Storage BigQuery VPC network

5. Enable the API. Click "Enable APIs & Services" from the sidebar.

API APIs & Services APIs & Services

Enabled APIs & services  
Library  
Credentials  
OAuth consent screen  
Page usage agreements

Traffic

▲ N

6. Click "+ Enable APIs and Services" at the top of the page.

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

Enabled APIs & services

1 hour 6 hours

7. Locate "Gmail API" on the API Library page and click on it. Then, click the Enable button.

8. Now configure your OAuth Consent Screen. If this is your first API project, Google may prompt you. Either way click "OAuth Consent Screen".

The screenshot shows the Google Cloud API console interface. On the left, a sidebar menu under the heading 'API APIs & Services' contains the following items: 'Enabled APIs & services', 'Library', 'Credentials', 'OAuth consent screen' (which is highlighted in blue), and 'Page usage agreements'. The main content area is titled 'OAuth consent screen' and displays the configuration for an application named 'MyWhistleBox'. There is an 'EDIT APP' link next to the application name. Below this, the 'Publishing status' is shown with a help icon. Under the 'Testing' section, there is a 'PUBLISH APP' button. The 'User type' section is currently set to 'External' with a help icon, and there is a 'MAKE INTERNAL' button below it.

9. On the "OAuth consent screen" page, select "**Make Internal**" or the Internal radio button as the user type. Click "Create".

**Note:** If you do not have a Google Workspace account, you must choose "External". We don't guarantee plain gmail accounts.

10. In Step 1, fill out the App Name (e.g. MyWhistleBox), User Support Email and Developer Contact email at the bottom. They can both be the same email. Then Save and Continue.

Google Cloud MyWhistleBox Server Search (/) for resources, d

**API** APIs & Services

- Enabled APIs & services
- Library
- Credentials
- OAuth consent screen**
- Page usage agreements

### Edit app registration

1 **OAuth consent screen** — 2 Scopes — 3 Test users — 4 Summary

#### App information

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

**App name \***  
MyWhistleBox  
The name of the app asking for consent

**User support email \***  
eradin1@gmail.com  
For users to contact you with questions about their consent. [Learn more](#)

11. On the "Scopes" page, click "Add or Remove Scopes".


✓ OAuth consent screen — 2 **Scopes** — 3 Test users — 4 Summary

Scopes express the permissions you request users to authorize for your app and allow your project to access specific types of private user data from their Google Account. [Learn more](#)

**ADD OR REMOVE SCOPES**

12. Then locate under Gmail API, "Send Email on your Behalf" in the list and select (hint, it's usually on the last page), then click Update.

<input type="checkbox"/>	API ↑	Scope	User-facing description
<input type="checkbox"/>	Gmail API	.../auth/gmail.addons.current.message.readonly	View your email messages when the add-on is running
<input checked="" type="checkbox"/>	Gmail API	.../auth/gmail.send	Send email on your behalf
<input type="checkbox"/>	Gmail API	.../auth/gmail.labels	See and edit your email labels
<input type="checkbox"/>	Gmail API	.../auth/gmail.settings.basic	See, edit, create, or change your email settings and filters in Gmail
<input type="checkbox"/>	Gmail API	.../auth/gmail.settings.sharing	Manage your sensitive mail settings, including who can manage your mail
<input type="checkbox"/>	Service	.../auth/service.management	Manage your Google API service configuration

Scopes express the permissions you request users to authorize for your app and allow your project to access specific types of private user data from their Google Account. [Learn more](#) 


[ADD OR REMOVE SCOPES](#)

## Your non-sensitive scopes

API ↑	Scope	User-facing description
No rows to display		

## Your sensitive scopes

Sensitive scopes are scopes that request access to private user data.

API ↑	Scope	User-facing description	
Gmail API	.../auth/gmail.send	Send email on your behalf	

## Your restricted scopes

Restricted scopes are scopes that request access to highly sensitive user data.

API ↑	Scope	User-facing description
No rows to display		

Then Click “**Save and Continue**”

13. Now create credentials. Click "**Credentials**" on the left side menu, click "+ Create Credentials", and select "**OAuth client ID**".

The screenshot shows the Google Cloud API & Services console. On the left, the 'APIs & Services' sidebar has 'Credentials' selected. The main content area is titled 'Credentials' and includes a '+ CREATE CREDENTIALS' button and a 'DELETE' button. Below this, there are three sections: 'API key', 'API Keys', and 'OAuth 2.0 Client ID'. The 'API key' section has a description: 'Identifies your project using a simple API key to check quota and access'. The 'API Keys' section is currently empty, showing 'No API keys to display'. The 'OAuth 2.0 Client ID' section is also empty, showing 'No OAuth clients to display'. A dropdown menu is open from the '+ CREATE CREDENTIALS' button, listing three options: 'API key', 'OAuth client ID' (which is highlighted), and 'Service account'. The 'Service account' option has a description: 'Enables server-to-server, app-level authentication using robot accounts'. Below the dropdown, there is a 'Help me choose' link with the text 'Asks a few questions to help you decide which type of credential to use'. At the bottom of the console, there are two empty tables. The first table is for 'API Keys' with columns for 'Name', 'Creation date', 'Type', 'Client ID', and 'Actions'. The second table is for 'OAuth 2.0 Client ID' with the same columns.

14. Select "**Web application**" on the "**Create OAuth client ID**" screen, then enter or copy and paste the Authorized Redirect URI:

<https://test.mywhistlebox.com/service-links/oauth/refresh.php>.

Name \*

MyWhistleBox App

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.



The domains of the URIs you add below will be automatically added to your [OAuth consent screen](#) as [authorized domains](#) [↗](#).

## Authorized JavaScript origins

For use with requests from a browser

[+ ADD URI](#)

## Authorized redirect URIs

For use with requests from a web server

URIs 1 \*

<https://mywhistlebox.com/service-links/oauth/refresh.php>




[+ ADD URI](#)


15. Click "Create" to generate your unique credentials. The **Client ID** and **Client Secret** information will display in a pop-up window. **Record both of these values so that you can enter them into MyWhistleBox email setup.**





## OAuth client created

The client ID and secret can always be accessed from Credentials in APIs & Services

 OAuth access is restricted to the [test users](#) listed on your [OAuth consent screen](#)

Your Client ID  

Your Client Secret  

 DOWNLOAD JSON

OK

16. Go to the next section with your Client ID and Client Secret and go to the next section to configure the Connector.

## Configuring MyWhistleBox Email Settings

Once the Google API has been configured, you can then configure MyWhistleBox's email connector using MyWhistleBox's Email Settings Page. You will need your Client Id, Client Secret and an email address that is part of your Workspace domain. Authorization is required. After saving the credentials, you should be able to successfully Authorize and send a Test Email.