

[Home](#) / [Dev guide](#) / [Provider guides](#) / [Google](#) / [Create a Google auth app](#)

Create a Google auth app

This page explains how to create and configure a Google Cloud Platform (GCP) application to use with your Nylas project.

Before you begin [↗](#)

Before you create your GCP application, you need to plan a couple fundamental parts of your project:

- [Choose your authentication method.](#)
- [Decide whether the application will be external or internal.](#)

Choose authentication method [↗](#)

First, you need to decide which authentication method works for you: Hosted OAuth or Custom authentication.



If you want to switch authentication methods later, you'll need to create and set up a new Google provider auth app. Your users will also have to re-authenticate with the new app.

Hosted OAuth is the fastest way to get started. If you're not interested in customizing your application, or you want to test with a few users, Nylas recommends you use Hosted OAuth.

Custom authentication lets you customize your application's auth process. This means your users will see your company name instead of "Nylas" on the OAuth screen. If you choose to use Custom auth, you must have an existing Nylas application and a callback URI.

Choose external or internal application [↗](#)

You also need to decide if you want to make your GCP application available to anyone (external) or only users that are part of your organization (internal).

If your GCP app needs to go through Google's security verification process, create an **external application**. This option allows users who aren't from your organization to authenticate with your



application" warning.

Google limits unverified external GCP applications to 100 authenticated accounts. To raise this limit, you need to complete Google's security verification process. For more information, see Google's official [Unverified apps documentation](#).

If you're creating a development or production app for internal use only, Nylas recommends you create an **internal application**. Only users who have accounts within your organization (for example, any user with an [@nylas.com](#) email address) will be able to access the application.

Internal GCP applications allow you to skip Google's verification and security review process. If anyone outside your organization needs to authenticate with your app, you'll need to go through Google's security review.

Create Google provider auth app [↗](#)



Nylas recommends you use separate GCP applications for your production and test environments.

Even small changes on a verified GCP app could trigger a new verification process. Having a separate app for your test environment gives you flexibility to test without interrupting your production users.

1. Go to the [Google Cloud Console Create Project page](#).
2. Give your project a **name**.
3. Select your project's **Organization** and **Location**.

New Project

Project name *

Sample Project One

?

Project ID: sample-project-one-327416. It cannot be changed later. [EDIT](#)

Organization *


sample.com

▼

?

Select an organization to attach it to a project. This selection can't be changed later.

Location *

 nylas.com

[BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

It might take several minutes for Google to create your project. When the process is finished, Google redirects you to the dashboard and displays a Create Project notification.

Notifications

✓

Create Project: Sample Project One

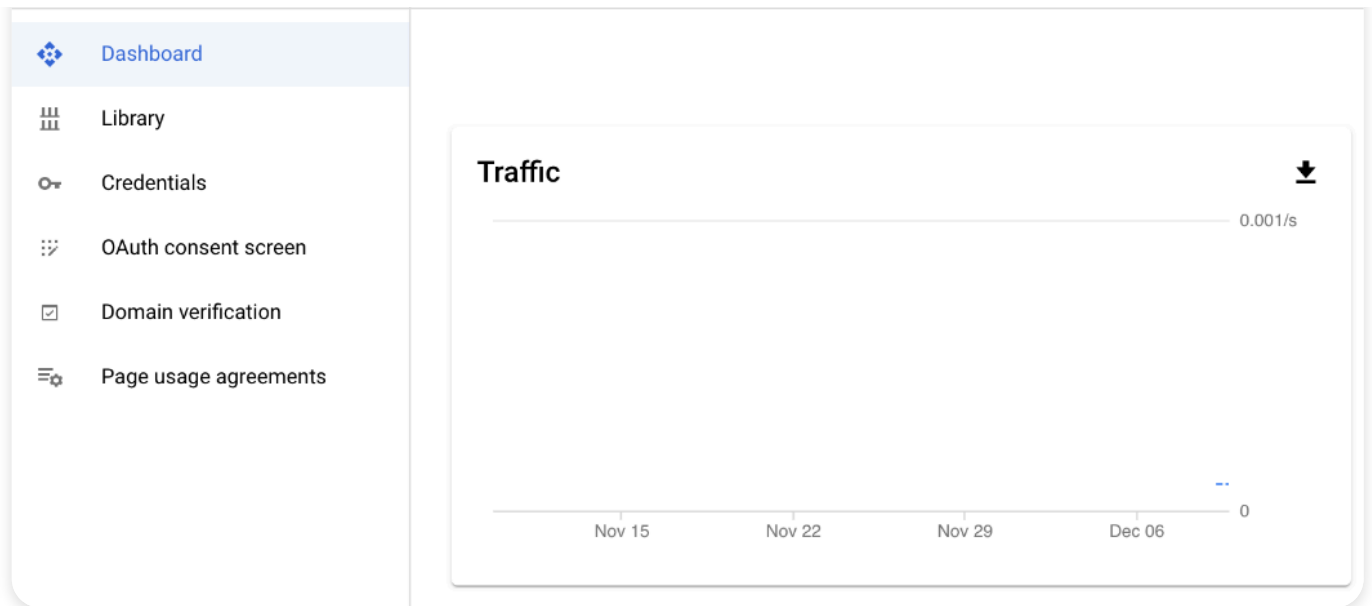
Just now

[SELECT PROJECT](#)

Enable required APIs [↗](#)

You need to enable certain APIs for your Google provider auth app to work with Nylas:

1. From the Google Cloud Platform dashboard, select **APIs and services**.
2. Click **Enable APIs and services**.



3. Search for and enable the following APIs:

- **Gmail API:** Required to read and send messages. Also required for the Threads, Drafts, Folders, and Files endpoints.
- **People API:** Required to use the Contacts endpoints.
- **Admin SDK API:** Optional. Grants access to room information for calendar events.

Google authentication scopes [↗](#)

You might need to take extra steps to comply with Google's OAuth 2.0 policies and complete their verification process before you can publish your GCP project.

Be sure to request the most restrictive [scopes](#) that you need for your project. If you request any of [Google's restricted scopes](#), Google will require your application to complete a security assessment. This could extend your verification timeline significantly, or cause Google to fail your review.

For more information, see Nylas' [Google verification and security assessment guide](#).



Nylas doesn't allow the all-access `mail.google.com` scope. This scope grants complete access to all Gmail features. Google automatically rejects verification for applications that include it, and makes you break down the access into individual, more granular scopes to complete verification.

Google scope URI	Description
https://www.googleapis.com/auth/userinfo.email	Required Google scope.
https://www.googleapis.com/auth/userinfo.profile	Required Google scope.

openid	Required Google scope.
https://www.googleapis.com/auth/gmail.modify	Read, compose, and send messages from a Gmail account.
https://www.googleapis.com/auth/gmail.readonly	View messages.
https://www.googleapis.com/auth/gmail.labels	View and edit Gmail labels.
https://www.googleapis.com/auth/gmail.compose	Create drafts and send messages.
https://www.googleapis.com/auth/gmail.send	Send messages.
https://www.googleapis.com/auth/calendar	View, create, edit, and delete calendars and events.
https://www.googleapis.com/auth/calendar.readonly	View calendars and events.
https://www.googleapis.com/auth/calendar.events	View and edit events on all calendars.
https://www.googleapis.com/auth/calendar.events.readonly	View events on all calendars.



If your GCP project uses the [gmail.readonly](#) or [gmail.labels](#) scopes, you need to [set up Pub/Sub](#). This ensures that you get real-time updates from your app.

Automatically include previously granted scopes [↗](#)

Nylas includes Google's [include_granted_scopes feature flag](#) when authenticating with Google OAuth 2.0. This feature flag tells Google to include any scopes that the user already approved on the specific GCP app (assuming the scopes are still valid). This simplifies the auth process for your users, because they're no longer required to re-select the scopes they already approved when they authenticate again.

Configure Google OAuth page [↗](#)

You can configure the OAuth page for both [internal](#) and [external](#) GCP applications. This is the page that your users are directed to when they authenticate with your Nylas application.

Configure internal OAuth page [↗](#)

1. From the Google Cloud Platform dashboard, select **OAuth consent screen**.



3. Fill out the required OAuth consent information and enter **nylas.com** as an **Authorized domain**.
4. Click **Save and continue**.
5. Select **Add or remove scopes**, and add the **.../auth/userinfo.email**, **.../auth/userinfo.profile**, and **openid** scopes.
6. Select the **scopes** needed for your application.
7. Review the **Summary** and ensure the information is correct.

Configure external OAuth page [↗](#)

1. From the Google Cloud Platform dashboard, select **OAuth consent screen**.
2. Choose the **External** user type and click **Create**.
3. Fill out the required OAuth consent information and enter **nylas.com** as an **Authorized domain**.
4. Click **Save and continue**.
5. Select **Add or remove scopes**, and add the **.../auth/userinfo.email**, **.../auth/userinfo.profile**, and **openid** scopes.
6. Select the **scopes** needed for your application.
7. Skip the **Test users** step for now.
8. Review the **Summary** and ensure the information is correct.
9. Click **Back to dashboard**.
10. Under **Publishing status**, click **Publish app**.

External App [EDIT APP](#)

Publishing status [?](#)

Testing

[PUBLISH APP](#)

User type

External [?](#)

[MAKE INTERNAL](#)

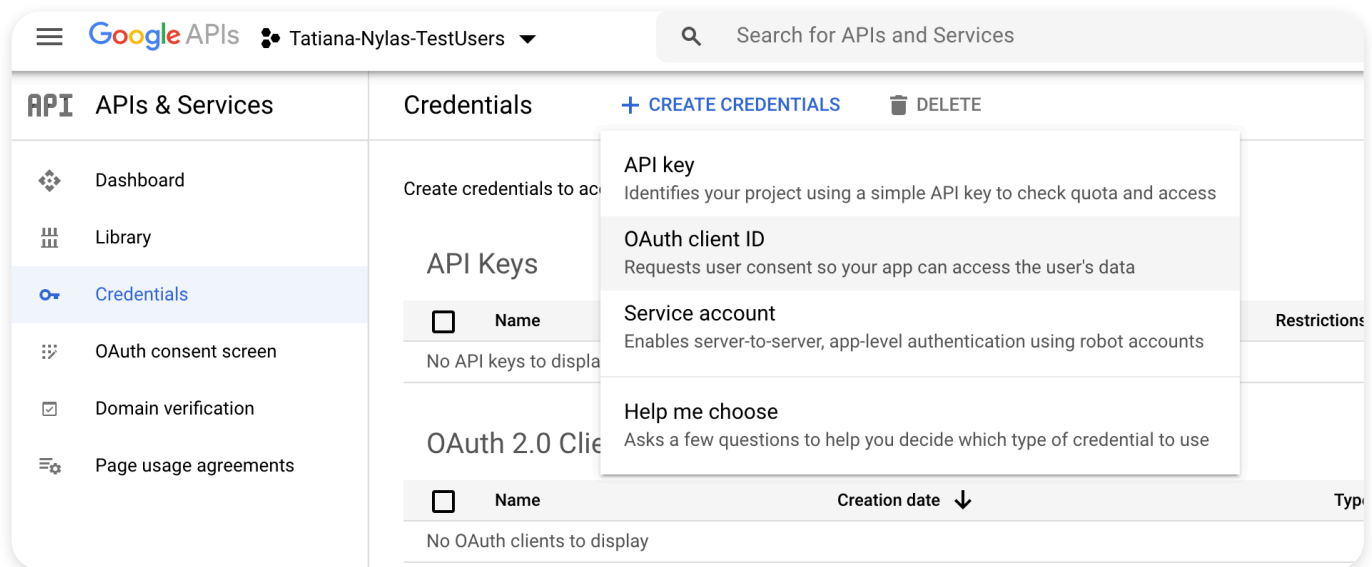


instead of adding them to Google individually as test users. The app is listed as unverified until you complete Google's [security review process](#).

Create Google application credentials [↗](#)

You need your GCP app's client ID and client secret to use the application with the Nylas APIs.

1. From the Google Cloud Platform dashboard, select **Credentials**.
2. Click **Create credentials** and choose **OAuth client ID** from the list.



3. Set the **Application type** to **Web application**.
4. Give the application a name.
5. Update the **Authorized redirect URIs**:
 - **U.S. Hosted auth**: <https://api.us.nylas.com/v3/connect/callback>
 - **E.U. Hosted auth**: <https://api.eu.nylas.com/v3/connect/callback>
 - **Custom auth**: Your project's callback URI.
6. Click **Create**. The client ID and secret are displayed in the **OAuth client created** notification.

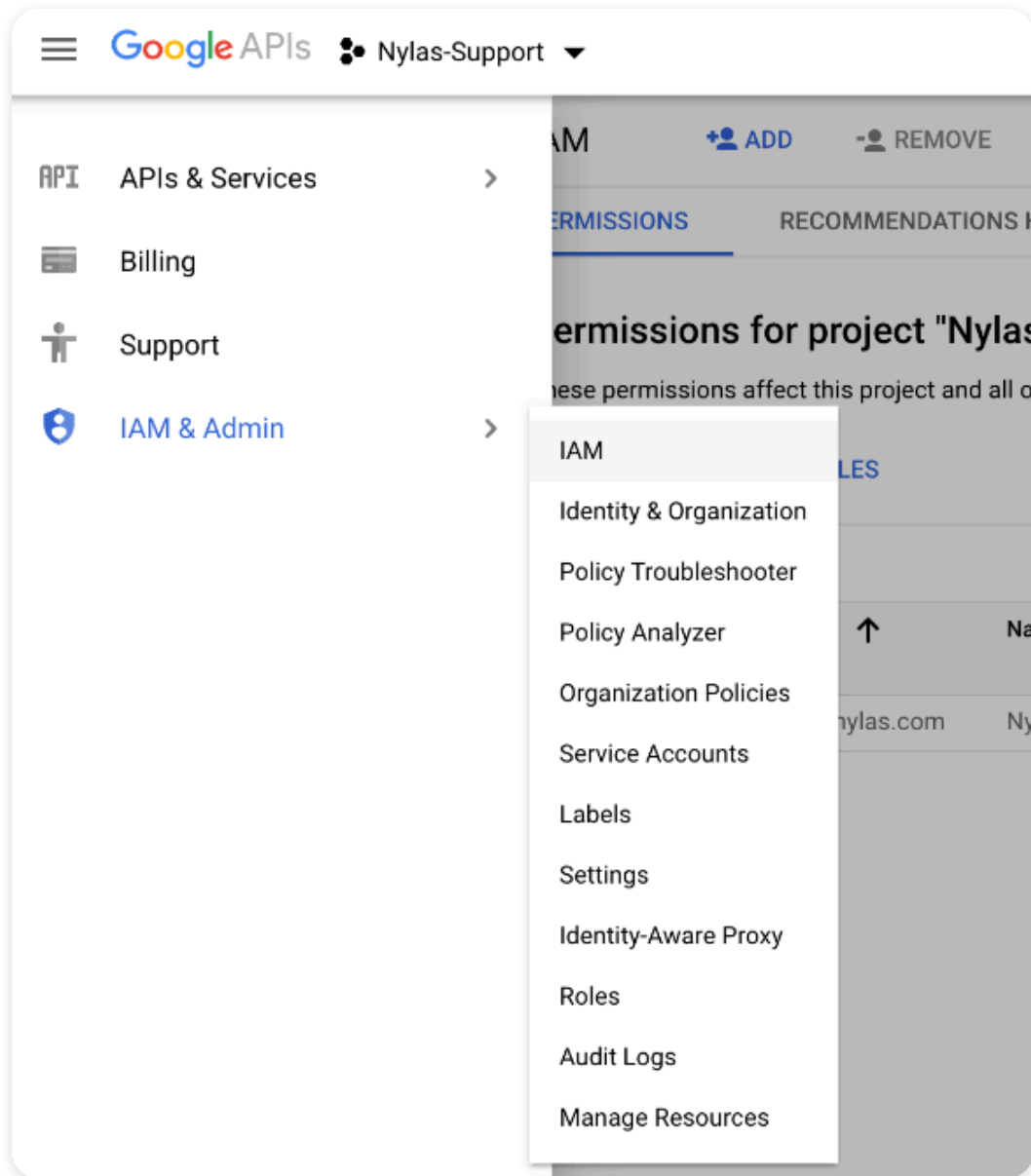


Be sure to save your client ID and secret somewhere safe, like a secrets manager. For best practices, see [Storing secrets securely](#).



Nylas recommends that you add the Nylas Support team to your GCP app as an application owner. This helps the team diagnose any issues that you might encounter.

1. From the Google Cloud Platform dashboard, open the navigation menu and select **IAM & admin > IAM**.



2. Click **Add**.
3. Add `support@nylas.com` as an owner.
4. Click **Save**.



Your GCP project needs to include a “Sign in with Google” button that meets [Google’s branding guidelines](#). This applies to the OAuth flow for both personal Gmail ([@gmail.com](#)) and Workspace email addresses.

For Hosted authentication, Nylas recommends you do one of the following:

- Configure the OAuth login prompt by setting the `prompt` parameter with `select_provider` or `detect,select_provider`. For more information, see [Configuring the OAuth login prompt](#).
 - If you add a `login_hint` that’s a personal Gmail or Workspace email address and you don’t configure a `prompt` during the Hosted auth flow, the user is immediately directed to the Google OAuth screen, without clicking the “Sign in with Google” button. This can result in delays or failure in verification.
- Use the pre-approved “Sign in with Google” button with the “Connect your account” button or other provider login buttons in your application. For more information, see Google’s official [Sign in with Google branding guidelines](#).

For Custom auth, use the pre-approved “Sign in with Google” button with the “Connect your account” button or other provider login buttons in your application.

For more information, see the [Google verification and security assessment guide](#).

Add a connector to your Nylas application [↗](#)



If you plan to use the Nylas Email API with Google, you need to [set up Google Pub/Sub](#) before you create a connector. If you don’t plan to use the Nylas Email API with your GCP app (for example, if you’re creating a calendar-only project), you can skip this step.

Your Nylas application communicates with external provider auth apps using [connectors](#). You can create a Google connector by copying the cURL request below and substituting your client ID, secret, and Pub/Sub topic name.



```
--header 'Authorization: Bearer <NYLAS_API_KEY>' \  
--header 'Content-Type: application/json' \  
--data '{  
  "name": "google example",  
  "provider": "google",  
  "settings":  
  {  
    "client_id": "<GCP_CLIENT_ID>",  
    "client_secret": "<GCP_CLIENT_SECRET>",  
    "topic_name": "<PUBSUB_TOPIC_NAME>"  
  },  
  "scope": [  
    "openid",  
    "https://www.googleapis.com/auth/userinfo.email",  
    "https://www.googleapis.com/auth/userinfo.profile",  
    "https://www.googleapis.com/auth/calendar",  
    "https://www.googleapis.com/auth/gmail.compose",  
    "https://www.googleapis.com/auth/gmail.modify"  
  ]  
}'
```

[Status](#)[Blog](#)[Support](#)[Roadmap](#)[Forums](#)[Cookies](#)[Trust Center](#)[Send Feedback](#)