

*Note: This document refers to Google-specific OAuth 2.0 settings. The end-to-end OAuth 2.0 process for MailIt is described in the **OAuth 2.0 General.pdf** document that references this document.*

Using Gmail IMAP OAUTH/XOAUTH2

The Gmail IMAP and SMTP servers have been extended to support authorization via the industry-standard OAuth 2.0 protocol. Using OAuth protocol, user can do authentication by Gmail Web OAuth instead of user and password. This way is more secure but also a little more complex.

Overview shared by Google:

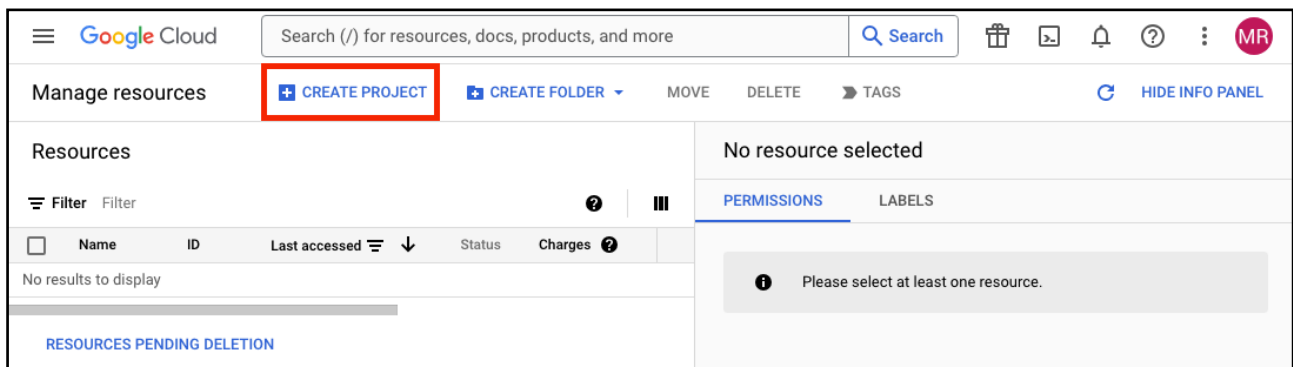
<https://developers.google.com/gmail/imap/imap-smtp?hl=en>

Create your project in Google Cloud

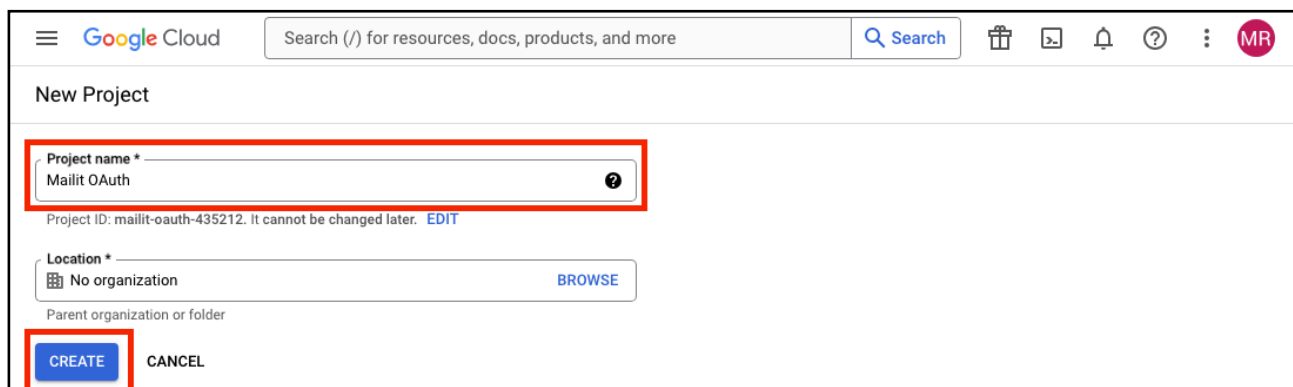
In order to use Gmail OAUTH authorization, you need to create an application in the Google Cloud. After creating the application, you will receive the Client ID and Client Secret, which must be used for authorization.

To create an application, go to: <https://console.cloud.google.com/cloud-resource-manager>

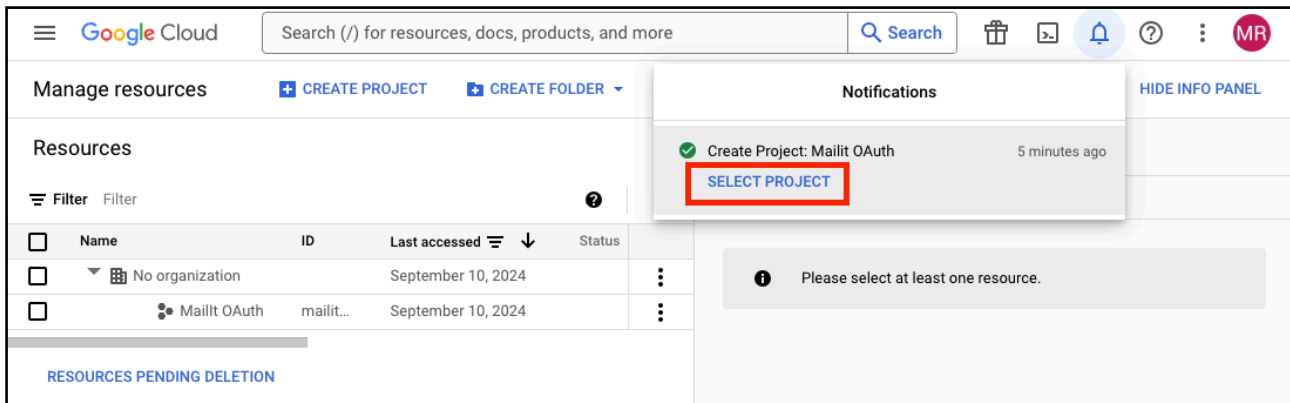
After logging in, you will be taken to the project management page in the Google Developers Console



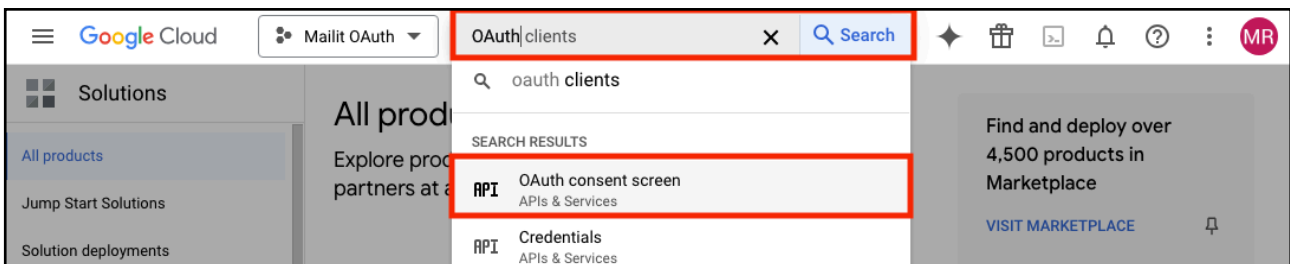
Click on the “CREATE PROJECT” button. You will be redirected to the page for creating a new project. Enter the name of your new project in the “Project name **” field and click the “CREATE” button. The project name “MailIt OAuth” is used as an example.



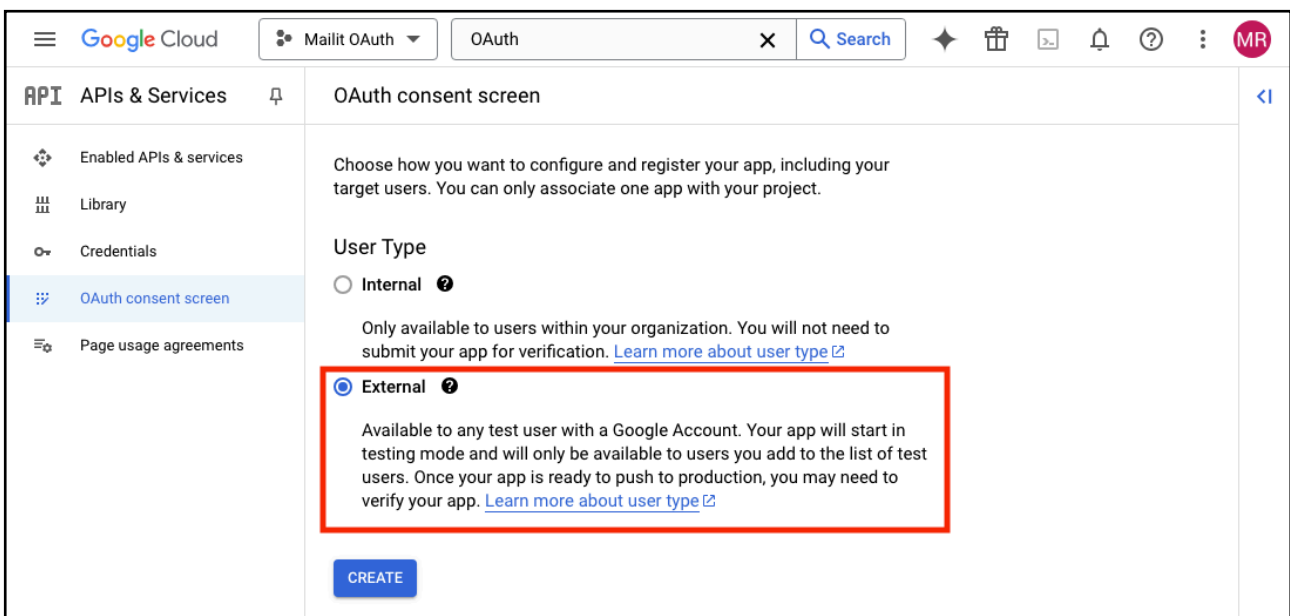
After clicking the “CREATE” button, you will be redirected to the project management page. The project creation process will begin, it will take a little time, after which you will see a notification that the project has been created with a project selection button. Click the Select Project button to select a project.



Next, you need to go to the “OAuth consent screen” section. To do this, start typing “OAuth consent screen” in the search bar. When the “OAuth consent screen” item appears in the search results, select it.



On the “OAuth consent screen” page, select the user type Internal or External. In the instructions, we use a personal account, so we will select External.



After you click the "Create" button, you will need to fill out a four-step form.

In the first step, fill in the required fields such as “App name” and User support email. Scroll down to the bottom of the page and enter your email in the “Developer contact information” field. Optionally, you can add a logo and fill in other non-mandatory fields. Then click the “Save and continue” button.

Google Cloud

Mailit OAuth

OAuth

Search

MR

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 *

Mailit OAuth

The name of the app asking for consent

User support email *

dacons.plugin.mailit@gmail.com

For users to contact you with questions about their consent. [Learn more](#)

App logo

This is your logo. It helps people recognize your app and is displayed on the OAuth consent screen.
After you upload a logo, you will need to submit your app for verification unless the app is configured for internal use only or has a publishing status of "Testing". [Learn more](#)

App logo preview



Logo file to upload

dacons mi logo 256.png

X

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.

Developer contact information

Email addresses *

dacons.plugin.mailit@gmail.com

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

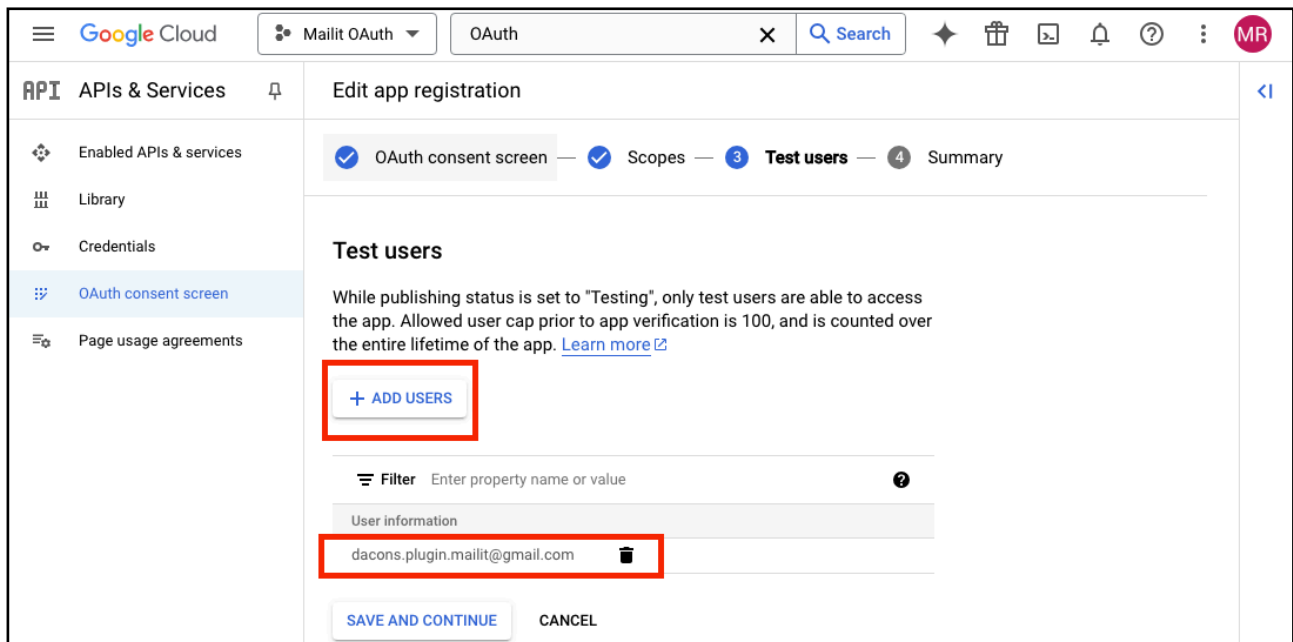
SAVE AND CONTINUE

CANCEL

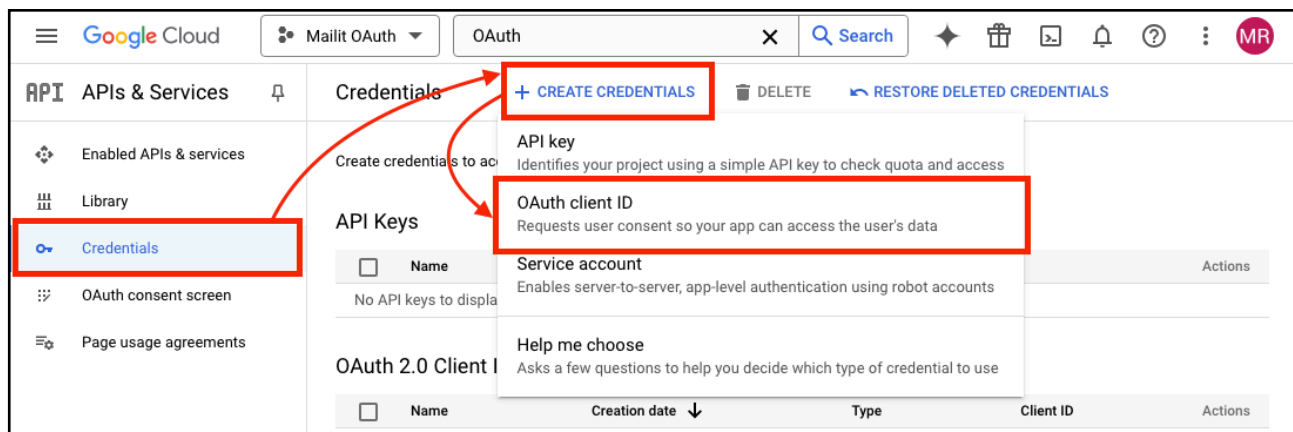
©2026 Dacons UG (haftungsbeschränkt)

In the second step “Scopes”, nothing else to do, go down to the bottom of the page and click the “Save and continue” button to go to the next step.

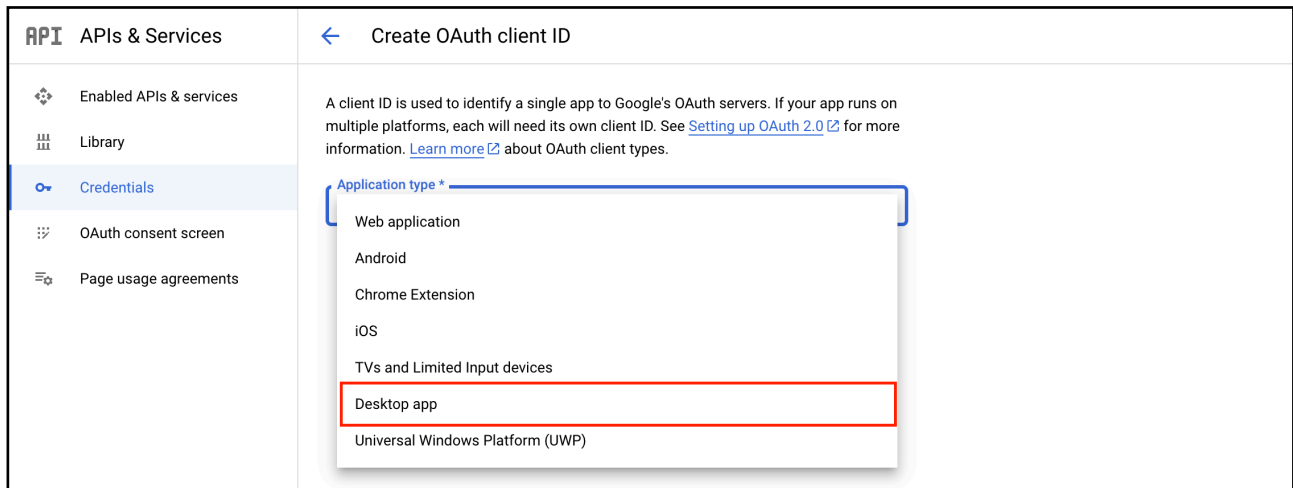
In the third step “Test users”, you will need to add your email with which you will pass OAuth authentication. Before proceeding to the next step, make sure the user is displayed in the list.



At the last step “Summary”, a summary of the application will be displayed, scroll down the page and click the “Back to dashboard” button. Now you can create a “Client ID” and “Client secret” for this, go to the “Credentials” section and click the “Create Credentials” button and select the “OAuth client ID” item in the drop-down menu.



In the next step, in the “Application type” field, select “Desktop app”.



APIs & Services

← Create OAuth client ID

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information. [Learn more](#) about OAuth client types.

Application type *

- Web application
- Android
- Chrome Extension
- iOS
- TVs and Limited Input devices
- Desktop app**
- Universal Windows Platform (UWP)

Enabled APIs & services

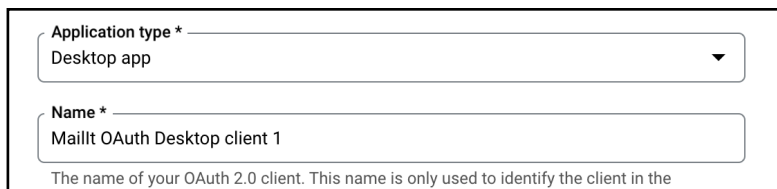
Library

Credentials

OAuth consent screen

Page usage agreements

After making a choice in the “Application type” field, the “Name” field will appear, in which you need to specify a name for the current “Client ID” / “Client secret” link. Specify any name by which you will be able to understand in the future for what purpose you created the credentials.

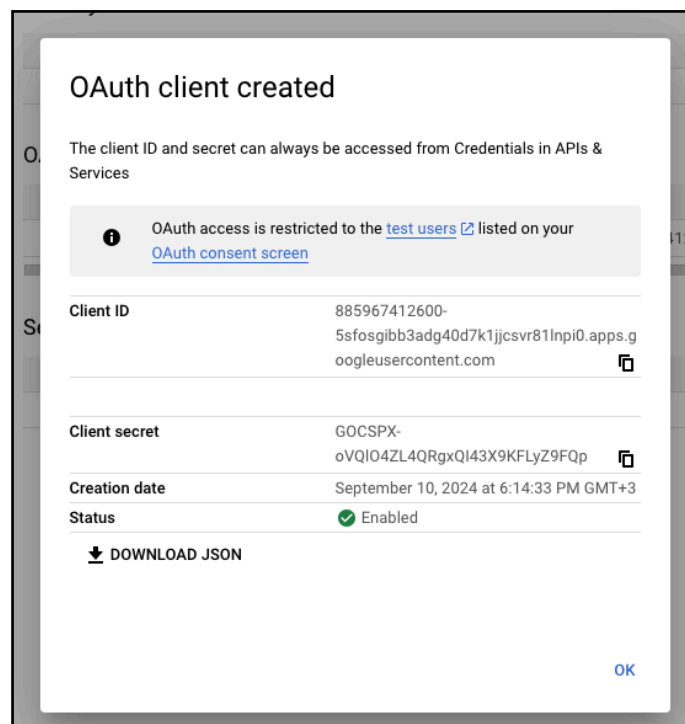


Application type *
Desktop app

Name *
MailIt OAuth Desktop client 1

The name of your OAuth 2.0 client. This name is only used to identify the client in the

After you fill in the "Name" field, click the "Create" button.
Next, a window will appear in which you can copy “Client ID” / “Client secret” and also download a JSON file in which there will be information about “Client ID” / “Client secret”.



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](#)

Client ID: 885967412600-5sfosglbb3adg40d7k1jcsvr81lnpi0.apps.googleusercontent.com

Client secret: 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

Creation date: September 10, 2024 at 6:14:33 PM GMT+3

Status: Enabled

DOWNLOAD JSON

OK

Now, please open the app **Simple Receiving IMAP.fmp12**, go to connection settings by clicking on the gear icon located at the top right and follow these steps:

1. In the “IMAP User Name” field, enter your email address.
2. In the field “Pref. IMAP Auth. ” select authorization type “OAUTH”.
3. Click on the “OAuth 2.0 Client Credentials” tab.
4. In the “OAuth Provider” drop-down list, select “gmail.com”.
5. Copy the Client ID and Client Secret from the browser into the corresponding fields “OAuth Client ID” and “OAuth Client Secret”.
6. Click the “Open OAuth2 in External browser” button to complete web authorization, as a result of which you can get a “Token”.

IMAP Connection Settings

IMAP Server

imap.gmail.com

IMAP Server Port

993

SSL/TLS

On

POP Server Timeout

60

IMAP Authentication only

Pref. IMAP Auth.

OAUTH

IMAP User Name

dacons.plugin.mailit@gmail.com

IMAP Password

Leave Messages on Server

On

Delete

Off

After (days)

7

Test Connection

Clear Settings

Last Session Transcript

OAuth 2.0 Client Credentials

OAuth 2.0 Additional settings

Log file

OAuth Provider

gmail.com

OAuth Client ID

885967412600-5sfosgibb3adg40d7k1jjcsvg81lnpi0.apps.googleusercontent.

OAuth Client Secret

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

Use Secret in Request

On

Open OAuth2 Window in FileMaker

Open OAuth2 in External browser

Hint: In order to use OAuth2 authorization you need to create an application.

For the GMail service:
<https://console.developers.google.com/>

For the Outlook or Live service:
<https://portal.azure.com/>

After creating the application, you will be available to "Client ID" and "Client Secret"

After clicking the “Open OAuth2 Window” button, a window with web authorization will open. Select your account from the list or add a new one if the account for which you created the application is not in the list.

 Sign in with Google

Sign in

to continue to [Mailit OAuth](#)

Email or phone

[Forgot email?](#)

[Create account](#)[Next](#)



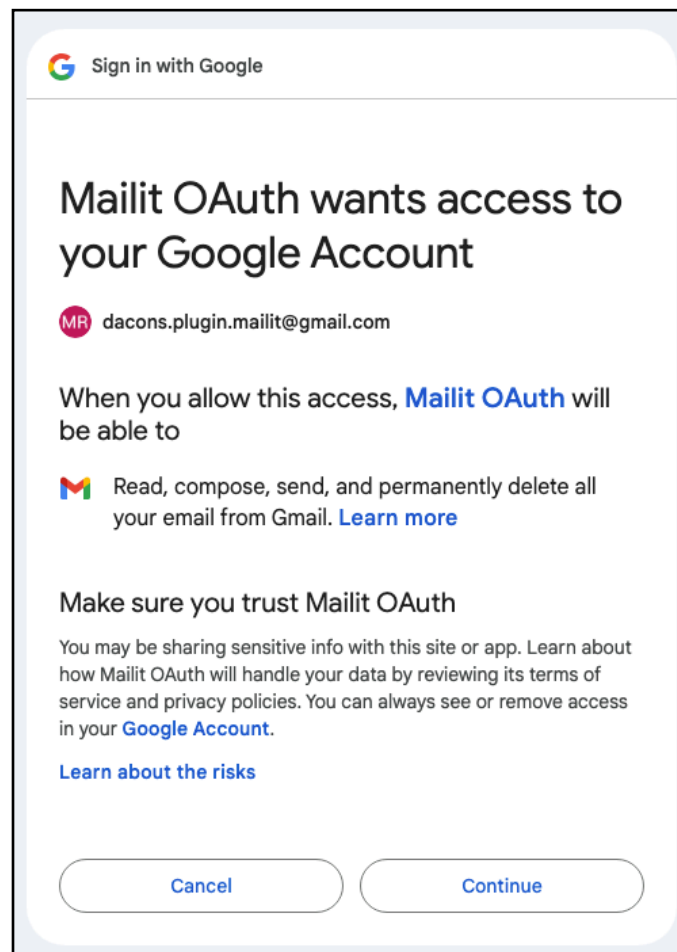
Google hasn't verified this app

You've been given access to an app that's currently being tested.
You should only continue if you know [the developer](#) that invited you.

[Continue](#)[Back to safety](#)

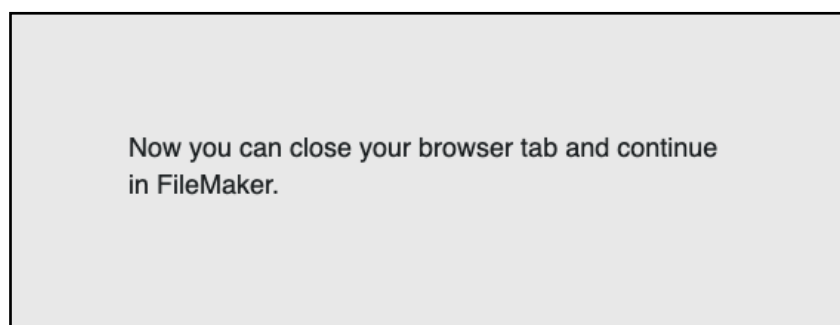
Since the application did not submit to Google's verification, you'll see a warning. Click on the "Continue" link, an additional warning will open with the option to continue by clicking the "Go to Mailit OAuth (unsafe)" link. Here, you will see a link with the name of your application.

On the next page, you need to allow the plugin access to your account by clicking the “Continue” button.

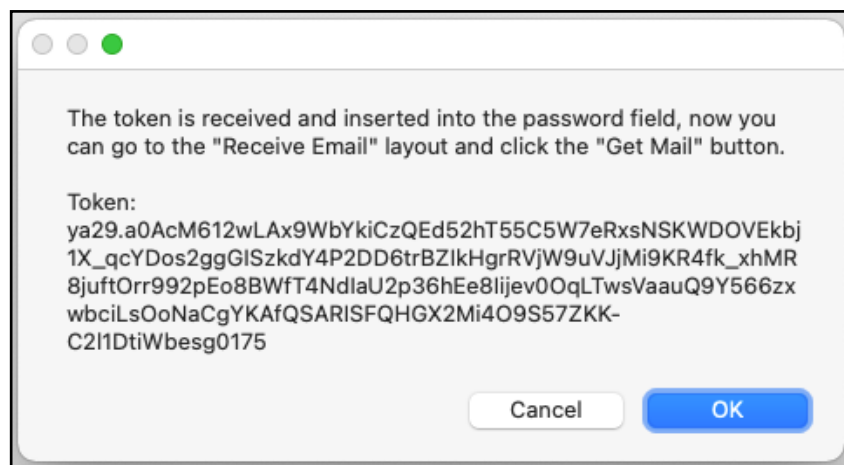


As a result, Google will pass the string “code” which is needed to get the “token”. If no errors occurred, then you should see the text in your browser: “Now you can close your browser tab and continue in FileMaker.”

If you clicked the “Open OAuth2 Window in FileMaker” button at the beginning, then click the “Get Code” button to pick up the line from the WebView and complete the request for a “token”.



Next, FileMaker will automatically execute a script that will exchange the code for a token. As a result of executing the script, you should see a message that the token has been received and inserted into the password field.



After receiving token, click the “Test Connection” button to test the connection.

In response to the “Token” request, two additional properties are transferred: “Refresh Token” and

A screenshot of a software interface with four tabs: "Last Session Transcript", "OAuth 2.0 Client Credentials", "OAuth 2.0 Additional settings" (which is selected), and "Log file". The selected tab contains a form with the following fields: "OAuth Refresh Token" with value "1//09BsBscMirbbICgYIARAAGAkSNwF-", "Authorization URL" with value "https://accounts.google.com/o/oauth2/v2/auth", "Token URL" with value "https://oauth2.googleapis.com/token", "Scope" with value "https://mail.google.com", and "Token Expires In" with value "Tuesday, 10.09.2024 19:33:14 EEST".

“Token Expires In”. “Token” is substituted in the password field. “Refresh Token” and “Token Expires In” can be found in the “OAuth 2.0 Additional settings” tab. In this example, the “Refresh Token” field displays the value in clear text so that authorization can be debugged, in real life, save the “Refresh Token” in a safe place.

The lifetime of “Token” is 1 hour from the moment it was received. After 1 hour, you need to request a new “Token” using “Refresh Token”. The implementation of the “Token” update process can be viewed in a script called “Test Connection” or “Connect to the mail server”.

On the “OAuth 2.0 Additional settings” tab, there are additional settings such as “Local WebServer Host” and “Local WebServer Port.” These fields are needed to configure the local web server. The local web server is needed to display the html file (../Mailit Assets/oauth/index.html) in FileMaker WebView, the file is accessible by address (http://localhost:1234/oauth/index.html) for web authorization. The local web server starts when the “Open OAuth2 Window” button is clicked and stops after clicking the “Get Code.” If for some reason the local server cannot start, then check that the plugin has permissions read / write to a local drive, and that the port specified in the settings “1234” is not being used by another application or service. If the port is busy. Try to specify any other port that is available on your system.

Additional Information

OAuth 2.0 Mechanism

<https://developers.google.com/gmail/imap/xoauth2-protocol>

Using OAuth 2.0 to Access Google APIs

<https://developers.google.com/identity/protocols/oauth2>