

GENESYS[®]

This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

E-mail Server Administration Guide

Setting up Gmail mailboxes for OAuth 2.0 authorization

5/13/2025

Setting up Gmail mailboxes for OAuth 2.0 authorization

For basic authentication, Genesys E-mail Server can access Gmail mailboxes using the IMAP, POP3, and SMTP protocols.

Starting with version 8.5.201.05, E-mail Server supports the OAuth 2.0 authorization access to Gmail mailboxes with the IMAP and SMTP protocols. Starting with version 8.5.202.02, OAuth 2.0 support is extended to POP3 and SMTP protocols.

To set up Gmail mailboxes using the OAuth 2.0 authorization access:

- 1. Create a Google application.
- 2. Configure Genesys E-mail Server.

Creating a Google application

For OAuth 2.0 authorization access to Gmail mailboxes with IMAP, POP3, and SMTP protocols, create a Google application in the Google platform. (In our example, the **esj** account is created.)

- 1. Follow this Google documentation to configure the application. The main configuration points are included in this procedure.
- 2. Select **Desktop** as an application type. E-mail Server uses "Manual copy/paste" as the redirect method.



- 3. Download the **client_id** and **client_secret** by clicking the Download arrow of your Google application. These values are required to get the access token and to configure Genesys E-mail Server.
- 4. On the OAuth consent screen, add your **App information** (App name, User support email). For example:

	Google Cloud Platform	Preparative Q. Search products and resources					
API	APIs & Services	Edit app registration					
♦ Ξ	Dashboard Library	OAuth consent screen — Scopes — Test users — Summary					
0+	Credentials	vientials					
5	CAUD: convent screen	App information					
8	Domain verification	This shows in the consent screen, and helps end users know who you are					
×.,	Page usage agreements	and contact you					
		App name * esj					
		The name of the app asking for consent					
		User support email * esjosufh2@gmail.com					
		For users to contact you with questions about their consent.					

(Click to expand)

5. In the Scope step, enter the Gmail scope as **Your restricted scopes**.

-	Google Cloud Platform	> mjoauth2gmail +	
RPI	APIs & Services	Edit app registration	
¢	Deshboard	API & Scope	User-facing description
=	Library	No rows to display	
0+	Credentials		
	GAuth consent screen	A Your sensitive sco	Des
	Domain verification	Sensitive scopes are scopes to	hat request access to private user data.
r_{0}	Page usage agreements	API + Scope No-scove to display	User facing description
		Your restricted scopes Restricted scopes Grnail scopes	pes that request access to highly sensitive user data.
		API 🕈 Scope	User-facing description
		https://mail .google .com/	Read, compose, send, and permanently-delete all your email from Gmail
(Cli	ck to expand)		

6. (Optional) Add test users. This step is for the testing phase. You can add an existing or new mailbox that the E-mail Server can access as a user. The application must be published after the testing phase before it can be used in production.

ur.T	APIs & Services	OAuth consent screen
÷	Deshboard	Test users
=	Library	While publishing status is set to "Testing", only test users are able to access
0+	Credentials	the app. Allowed user cap prior to app verification is 100, and is counted over the entire lifetime of the app. Learn more
y.	CAuth consent screen	+ ADD-USERS
8	Domain verification	
74	Page usage agreements	a users (3 test, 0 other) / 100 user cap
		T Filtertable
		A In order to limit abuse, users can be added, but not removed
		User information
		customer1.mjoauth2@gmail.com
		contractions and a second

- 7. Get a refresh token manually. Follow the steps as described in this Google documentation to get the OAuth 2.0 refresh token:
 - **Step 1**: Generate a code verifier and challenge. (Note: A refresh token can be acquired without this

step.)

• **Step 2**: Send a request to Google's OAuth 2.0 server. In a web browser, enter the following as a URL, replacing *<your client id>* with your application client ID:

```
https://accounts.google.com/o/oauth2/
auth?scope=https://mail.google.com/&redirect_uri=urn:ietf:wg:oauth:2.0:oob&response_type=code&clien
client id>
```

Note that **redirect_uri** and **response_type** values cannot be changed.

• **Step 3**: Google prompts the user for consent. You will be prompted to sign in with a mailbox if it was not signed in. In the test phase, if you have multiple mailboxes, only the mailboxes that have been added as Test Users can access the application. This may change after the application is published. After signing in with mailbox credentials, there will be an alert in the test phase. Click **Continue**:

C	Google hasn't ve	erified th	nis app
You'v teste that i	ve been given access to an d. You should only continue invited you.	app that's cu e if you know	rrently being the developer
Cont	tinue	Back	to safety
Click A	llow:		
Gr	ant esj permiss	sion	
Μ	Read, compose, s permanently dele email from Gmail	send, and ete all you	d µr ∽
		Deny	Allow

Google

	Confirm your choices
	esjoauth2@gmail.com
You	are allowing esj to:
~	Read, compose, send, and permanently delete all your email from Gmail
Mak	ke sure you trust esj
You r Learn term or re	may be sharing sensitive info with this site or app. n about how esj will handle your data by reviewing its is of service and privacy policies. You can always see move access in your Google Account .
Lear	n about the risks
Can	Allow
ie Aut	thorization Code is displayed. Copy the Code by clicking Copy Icor
	Google
	Sign in

• Step 4: Exchange authorization code for refresh and access tokens.

The Authorization Code acquired in Step 3 can be used to exchange for OAuth 2.0 access and refresh tokens within 10 minutes (after you received the authorization code) by means of the following command:

curl -d "code=<your authorization code>&grant_type=authorization_code&redirect_uri=urn:ietf:wg:oauth:2.0:oob&client_id=<your client_id>&client_secret=<your client secret>" -X POST https://oauth2.googleapis.com/token

Replace <*your authorization code*>, <*your client_id*>, and <*your client_secret*> with the actual values of your application. Keep the rest as is.

Here is an example of the response:

{

"expires_in": 3599,

"scope": "https://mail.google.com/",

"token_type": "Bearer"

}

The Bearer **access_token**, which can be used to access the mailbox in IMAP/POP3/SMTP, expires every 3600 seconds. The **refresh_token** can be used to get a new Bearer token. After the application is published, the refresh code will only expire under the conditions listed in this Google document.

Configuring Genesys E-mail Server

To configure E-mail Server:

- Set the JavaMail property mail.<type>.auth.mechanisms (where <type> can be imap, pop3, and smtp) to X0AUTH2. (To disable OAuth 2.0, remove the JavaMail property.)
- Configure options in the **smtp-client** section.
- Configure options in the **pop-client** section.
- Configure options in the **secret**-<**secretName**> section.

Configuring the **smtp-client** section

Configure the following configuration options:

- client-id—Specify the Client ID of the Google application.
- password—Specify the refresh_token of the SMTP account.
- secret—Specify the secretName of the secret-<secretName> section to associate with the Google application secret.
- tenant-authority—Specify the valid Google authority server, which is a case-insensitive string that contains "google".

Configuring the **pop-client** section

Configure the following configuration options:

- client-id—Specify the Client ID of the Google application.
- password—Specify the refresh_token of the Gmail mailbox.

- secret—Specify the secretName of the secret-<secretName> section to associate with the Google application secret.
- tenant-authority—Specify the valid Google authority server, which is a case-insensitive string that contains "google".

Configuring the **secret-**<**secretName**> section

• password—Specify the client_secret value of the registered Google account.

Limitations

- During the test phase, a refresh token expires in 7 days.
- During the test phase, the refresh token may stop working if the Gmail mailbox is not used for a few days.