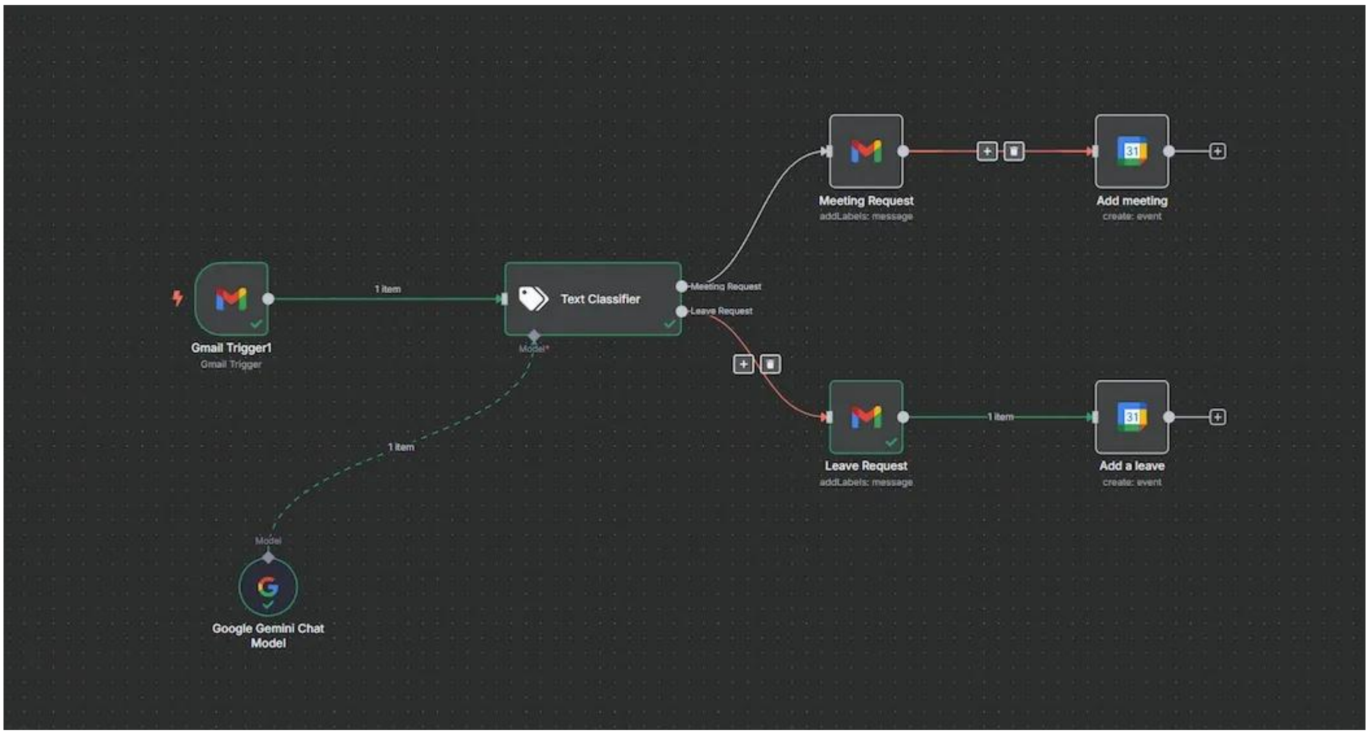
n8n is an open-source automation tool that integrates with a wide variety of services and APIs to help automate workflows. Using n8n, we can build an email classifier that automates the sorting of incoming emails based on the content such as scheduling of meeting or leave request. Lets build a n8n based project which labels the mails according to their category such as a meeting information or leave request and many more.



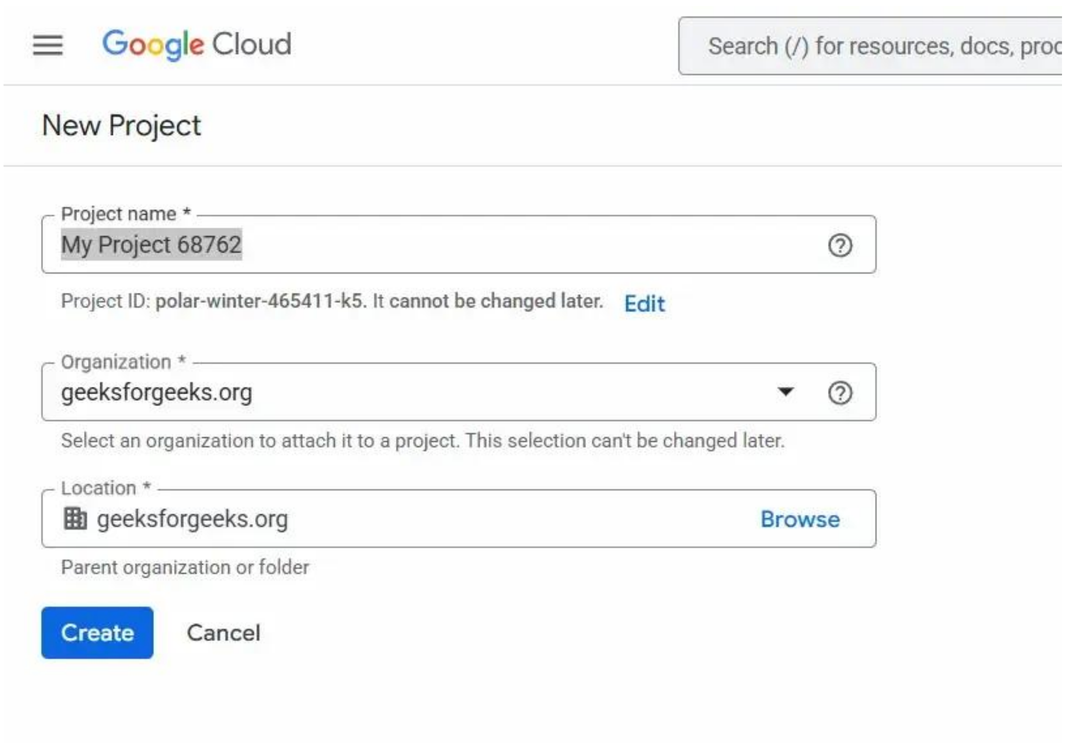
Overview of Email Classifier based on n8n

API Authentication

Firstly we need to get API Authentication for our Gmail API and Google Calendar API. Follow the steps to get them.

Step 1: Go to the official website of google cloud console and login.

Step 2: After successful login, create a New Project.



New Project

Project name *
My Project 68762

Project ID: polar-winter-465411-k5. It cannot be changed later. Edit

Organization *
geeksforgeeks.org

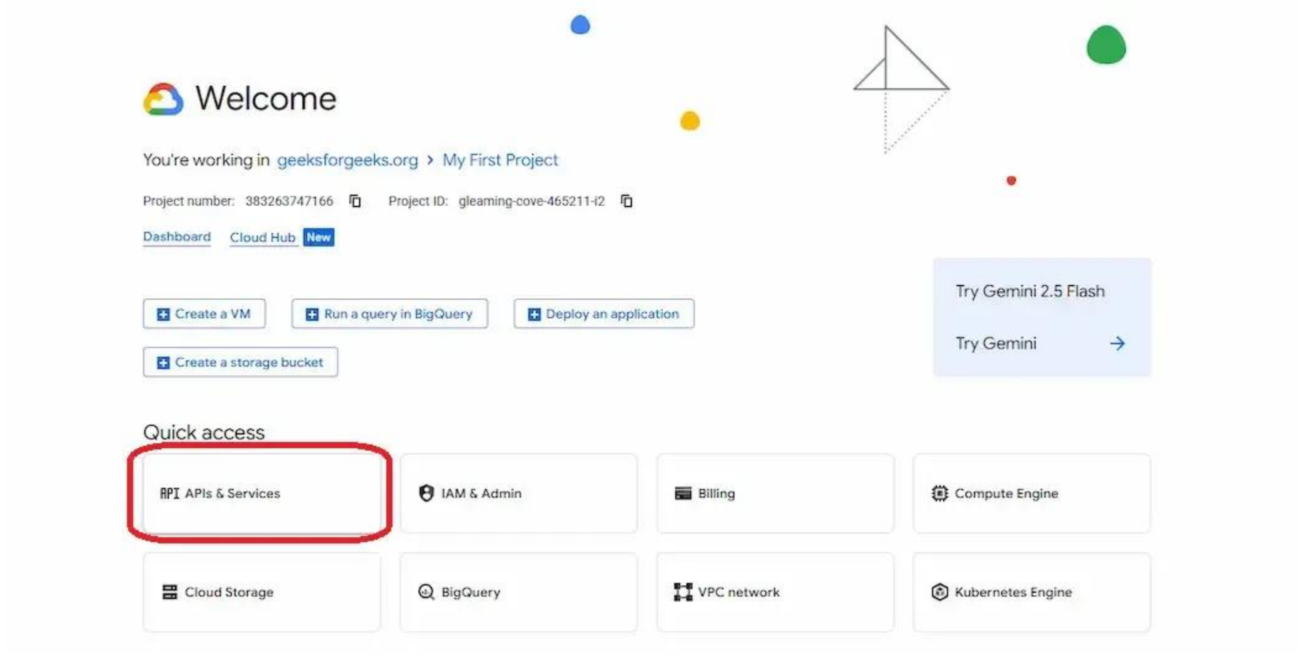
Select an organization to attach it to a project. This selection can't be changed later.

Location *
geeksforgeeks.org Browse

Parent organization or folder

New Project

Step 3: Find and enter the APIs & Services tab.

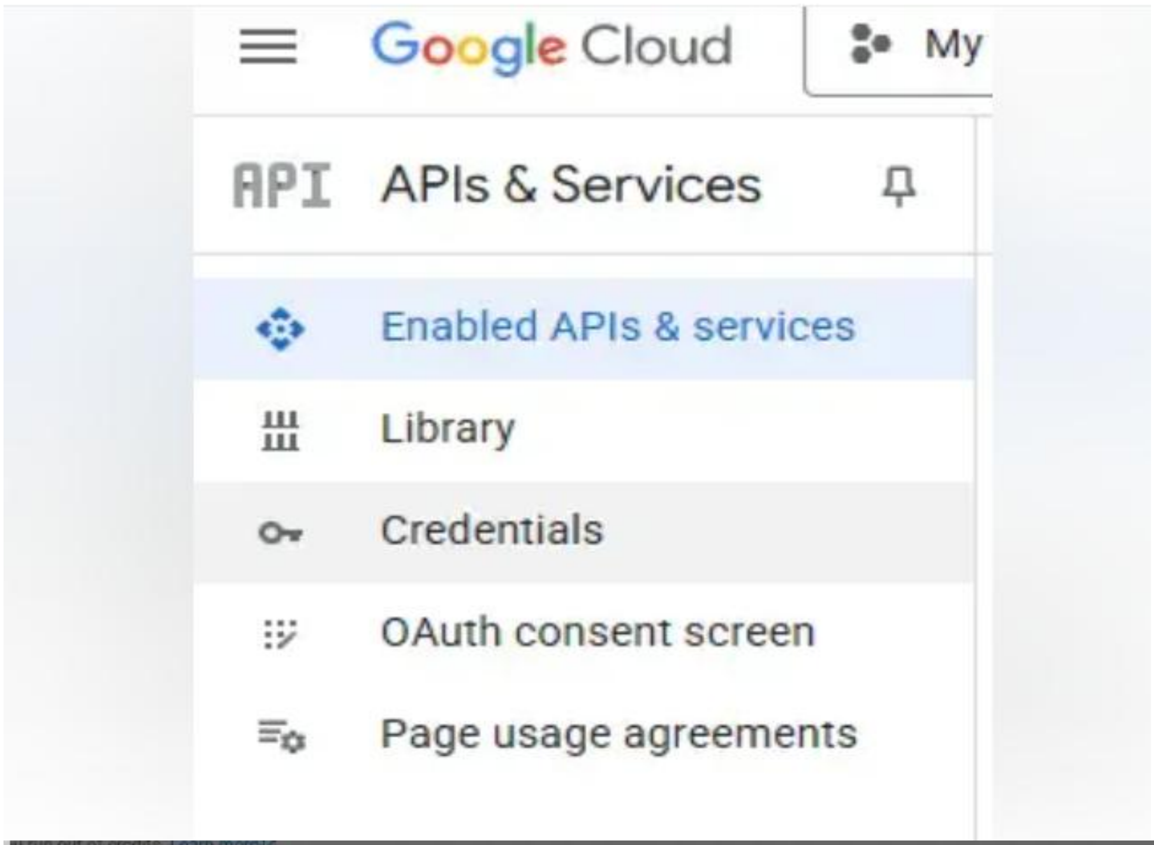

Welcome

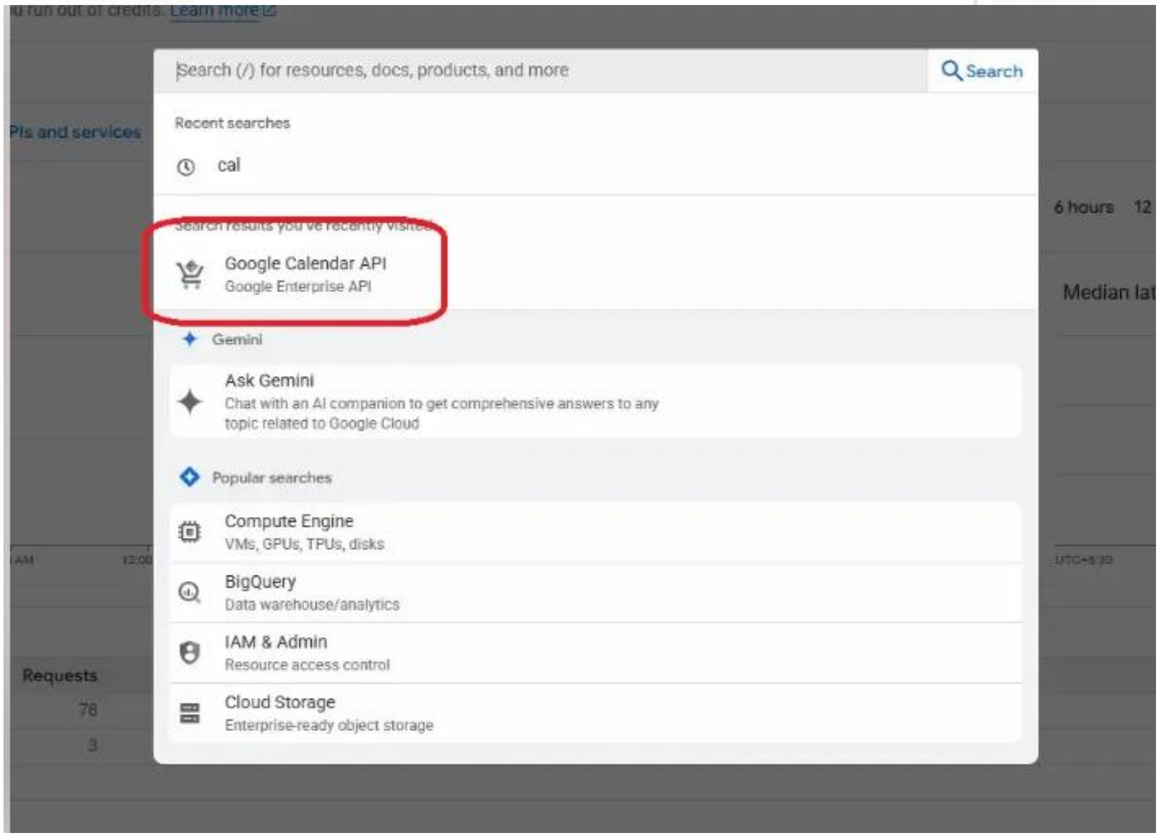
You're working in geeksforgeeks.org › My First Project

Project number: 383263747166 Project ID: gleaming-cove-465211-i2

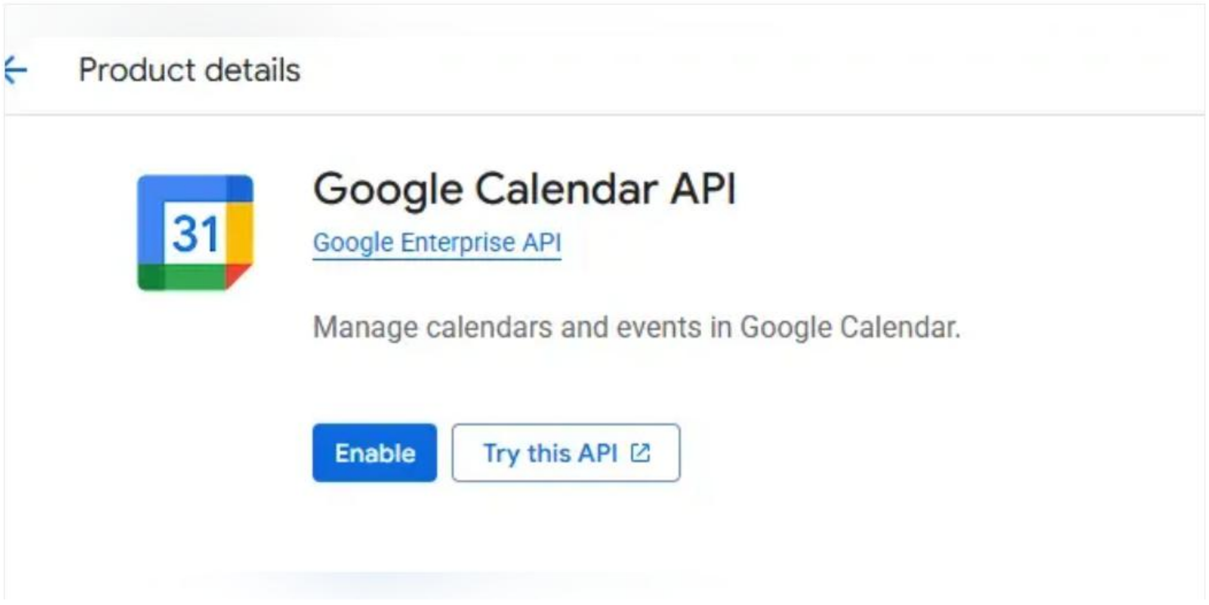
APIs & Services tab

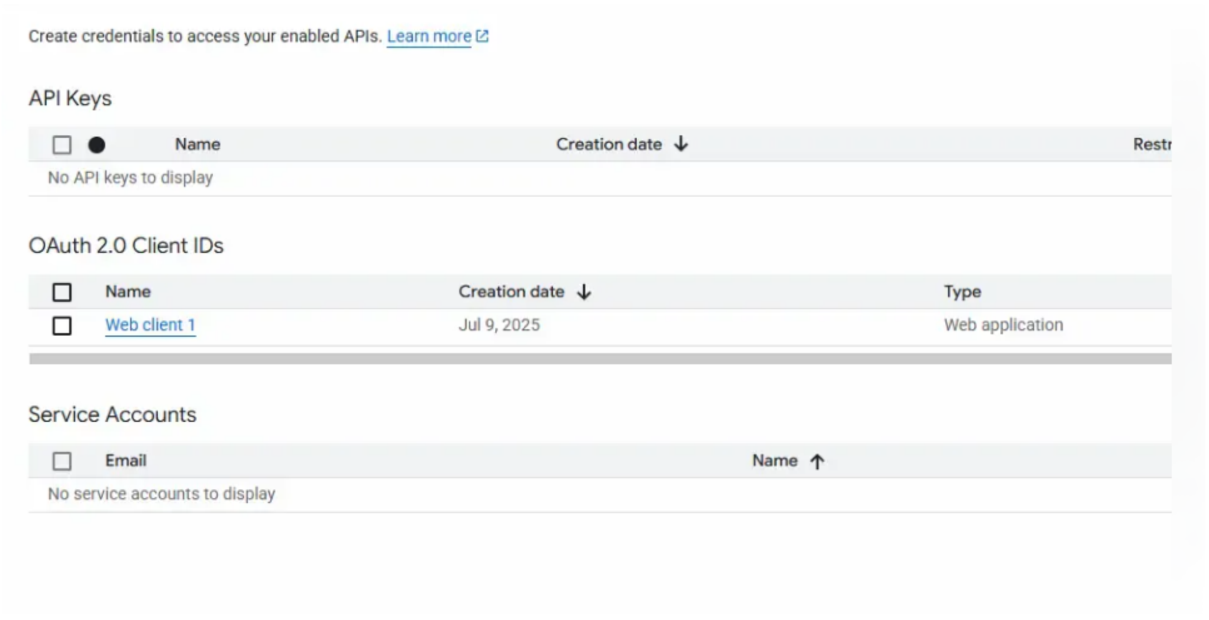
Step 4: From the menu, select **Enabled APIs & Services.** Search for Google Calendar API and Gmail API.


Google Cloud
My
APIs & Services
Enabled APIs & services
Library
Credentials
OAuth consent screen
Page usage agreements


Search (/) for resources, docs, products, and more
Search
Recent searches
cal
Search results you've recently visited
Google Calendar API
Google Enterprise API
Gemini
Ask Gemini
Chat with an AI companion to get comprehensive answers to any topic related to Google Cloud
Popular searches
Compute Engine
VMs, GPUs, TPUs, disks
BigQuery
Data warehouse/analytics
IAM & Admin
Resource access control
Cloud Storage
Enterprise-ready object storage

Step 5: Enable the APIs.




Create credentials to access your enabled APIs. Learn more

API Keys

Name	Creation date	Restr
No API keys to display		

OAuth 2.0 Client IDs

Name	Creation date	Type
Web client 1	Jul 9, 2025	Web application

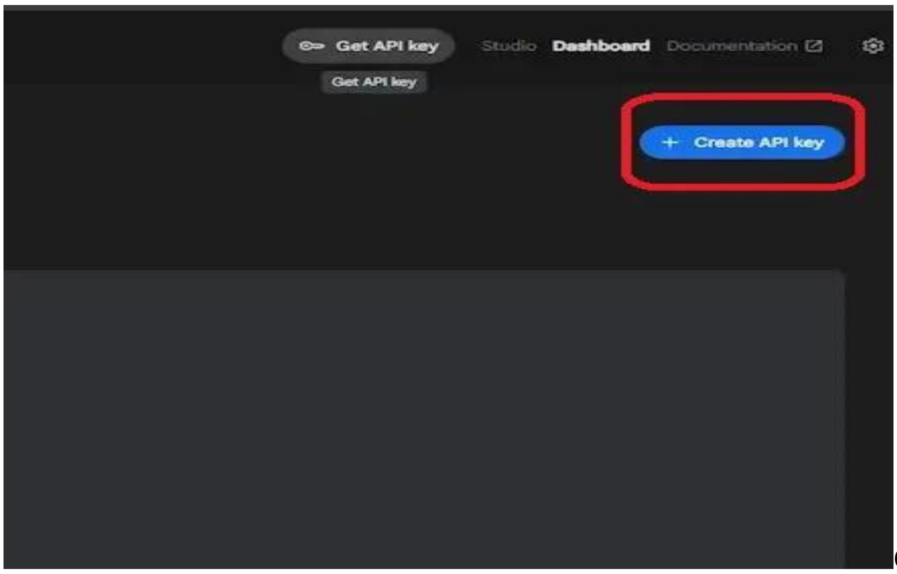
Service Accounts

Email	Name
No service accounts to display	

Steps to extract Google Gemini API Key

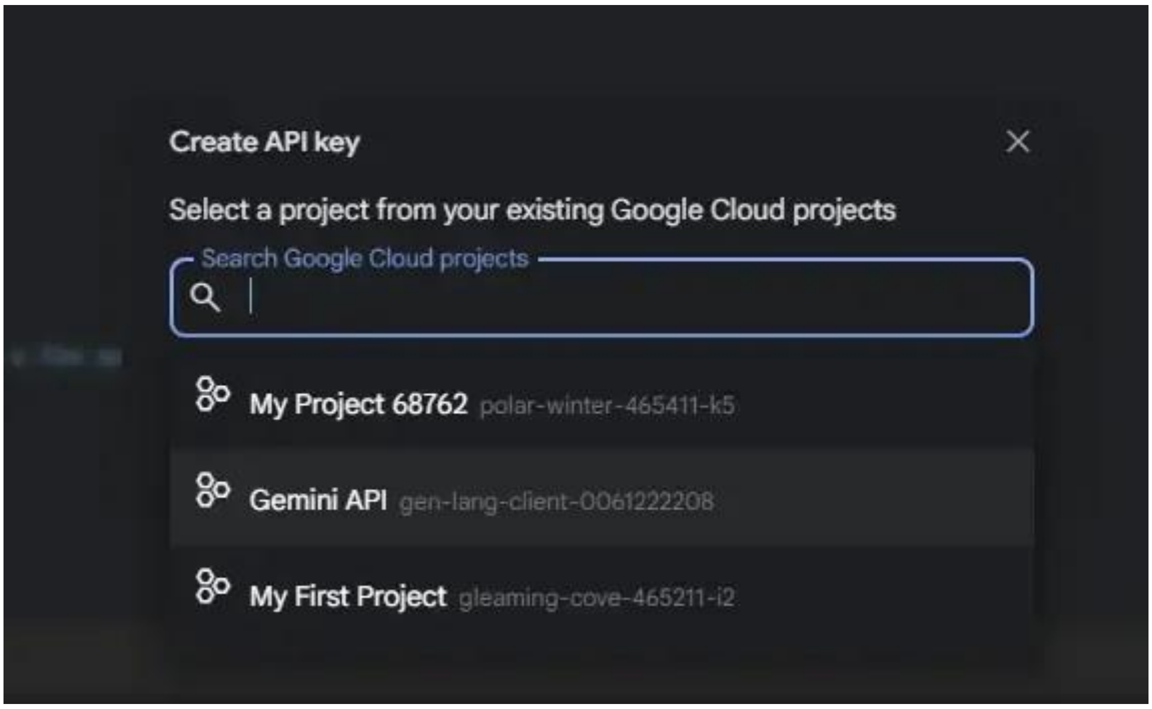
Step 1: Go to the official website of Google Gemini API.

Step 2: Click on Create New API Key.



Create API Key

Step 3: Select Your Project or Create a new project.


Create API key

Select a project from your existing Google Cloud projects

Search Google Cloud projects

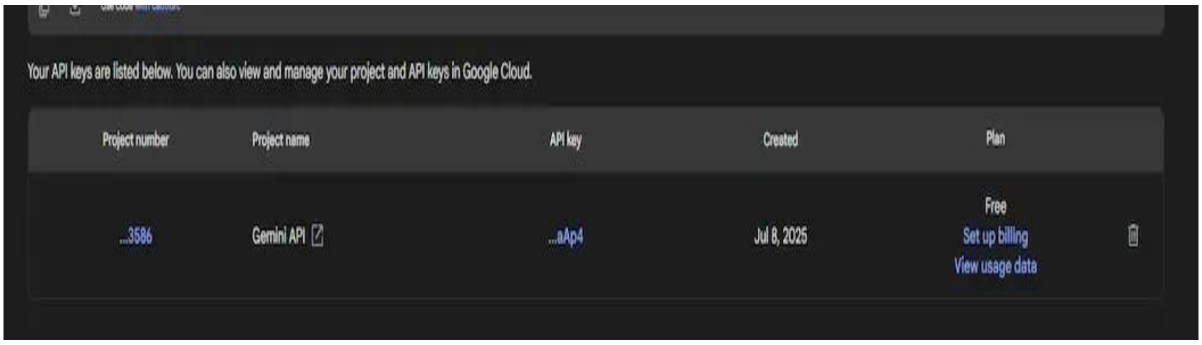
My Project 68762 polar-winter-465411-k5

Gemini API gen-lang-client-0061222208

My First Project gleaming-cove-465211-i2

Create a new project.

Step 4: Copy the created API Key and save it for use.


Your API keys are listed below. You can also view and manage your project and API keys in Google Cloud.

Project number	Project name	API key	Created	Plan
...3586	Gemini API	...aAp4	Jul 8, 2025	Free Set up billing View usage data

API

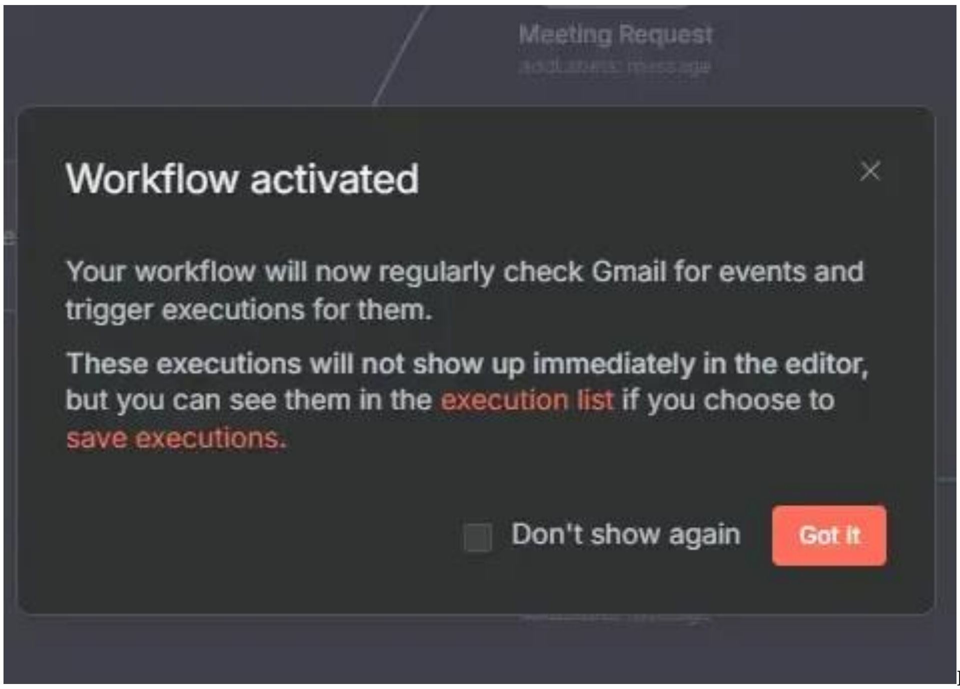
Workflow Creation on n8n

Follow the steps to build the email classifier agent.

For setting up your n8n refer to: What is n8n?

Creating Gmail Trigger

Step 1: Create a new workflow.



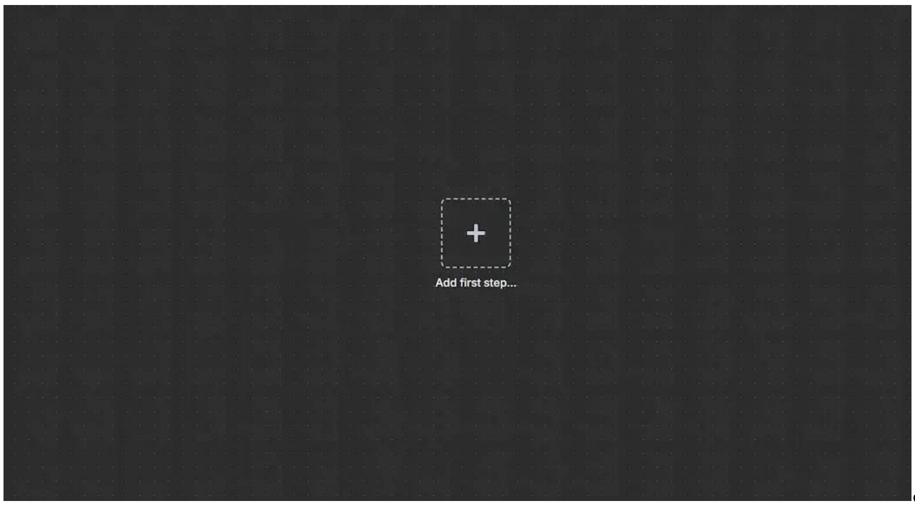
Workflow activated

Your workflow will now regularly check Gmail for events and trigger executions for them.

These executions will not show up immediately in the editor, but you can see them in the execution list if you choose to save executions.

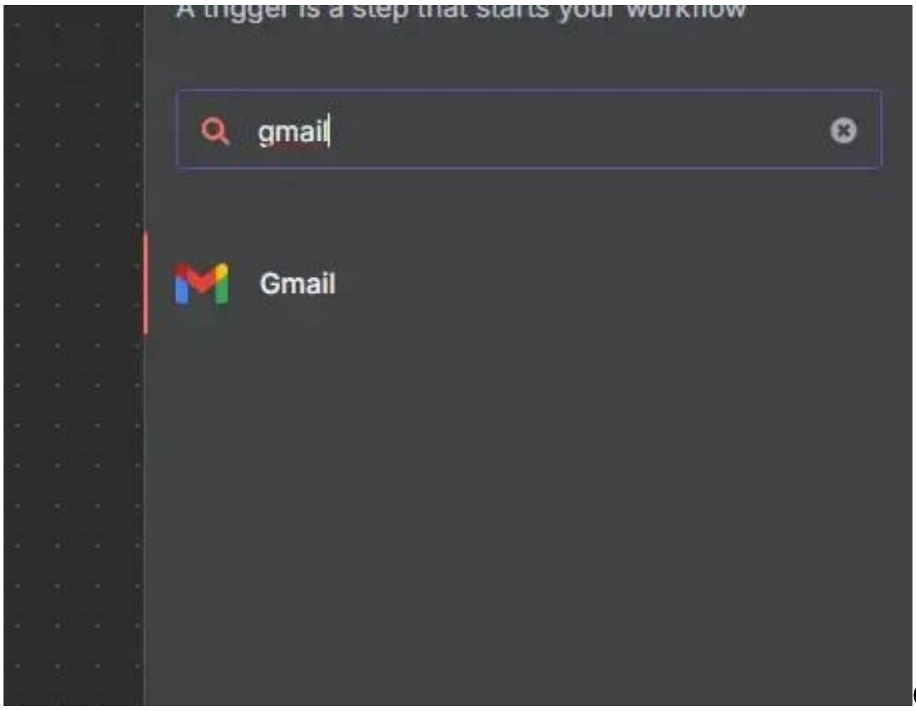
new workflow

Step 2: Click on add first step button.



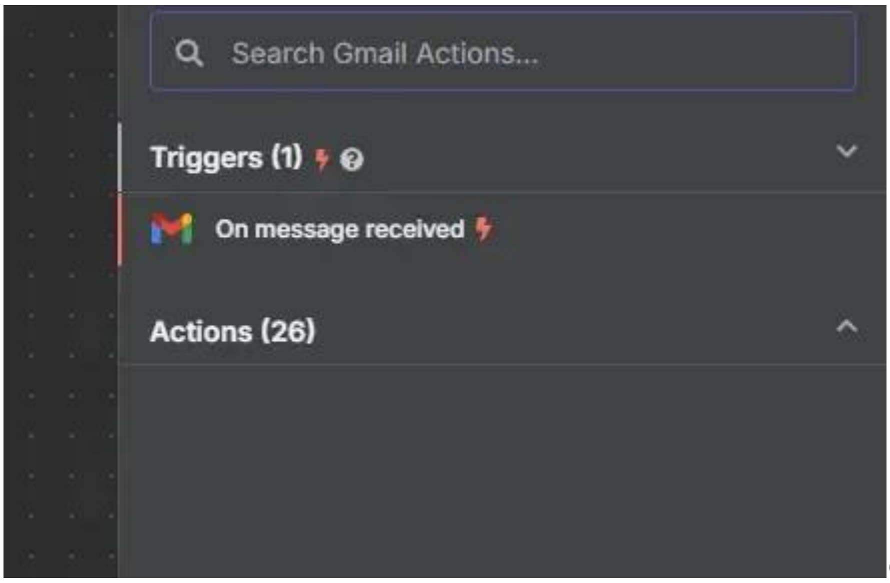
add first step button

Step 3: Search for Gmail.



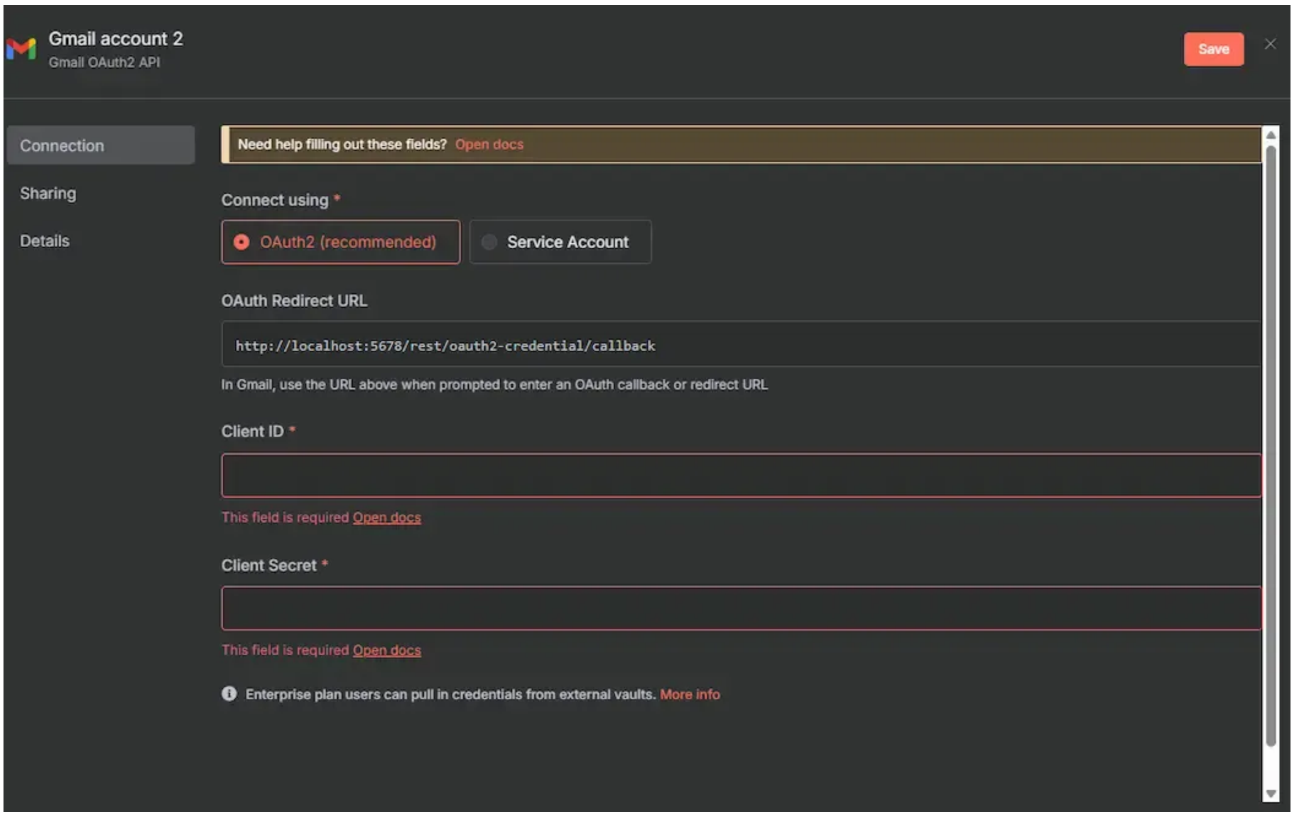
Gmail

Step 4: Select the **On message received** trigger.



On message received

Step 5: The Gmail trigger will open.



```
http://localhost:5678/rest/oauth2-credential/callback
```

In Gmail, use the URL above when prompted to enter an OAuth callback or redirect URL

Client ID *

This field is required Open docs

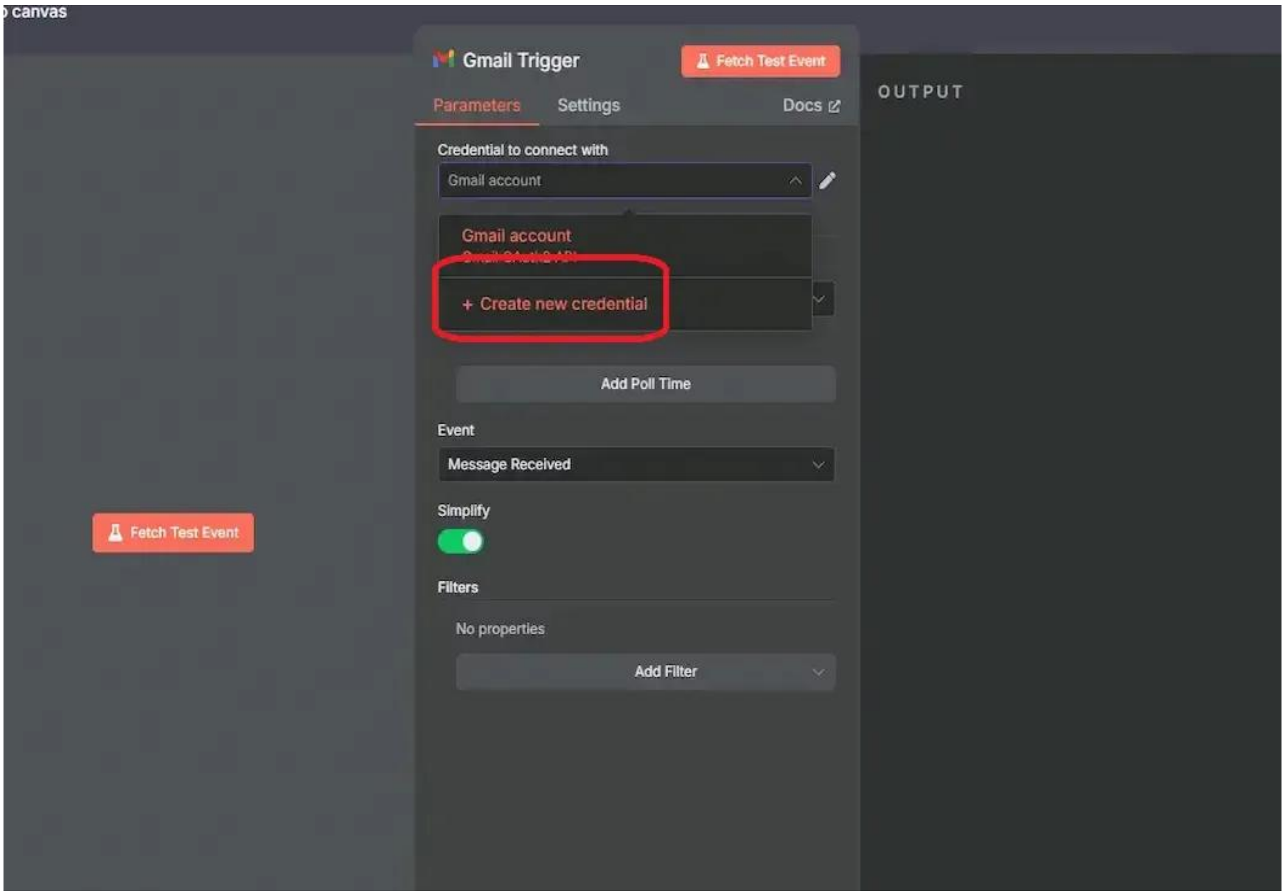
Client Secret *

This field is required Open docs

Enterprise plan users can pull in credentials from external vaults. More info

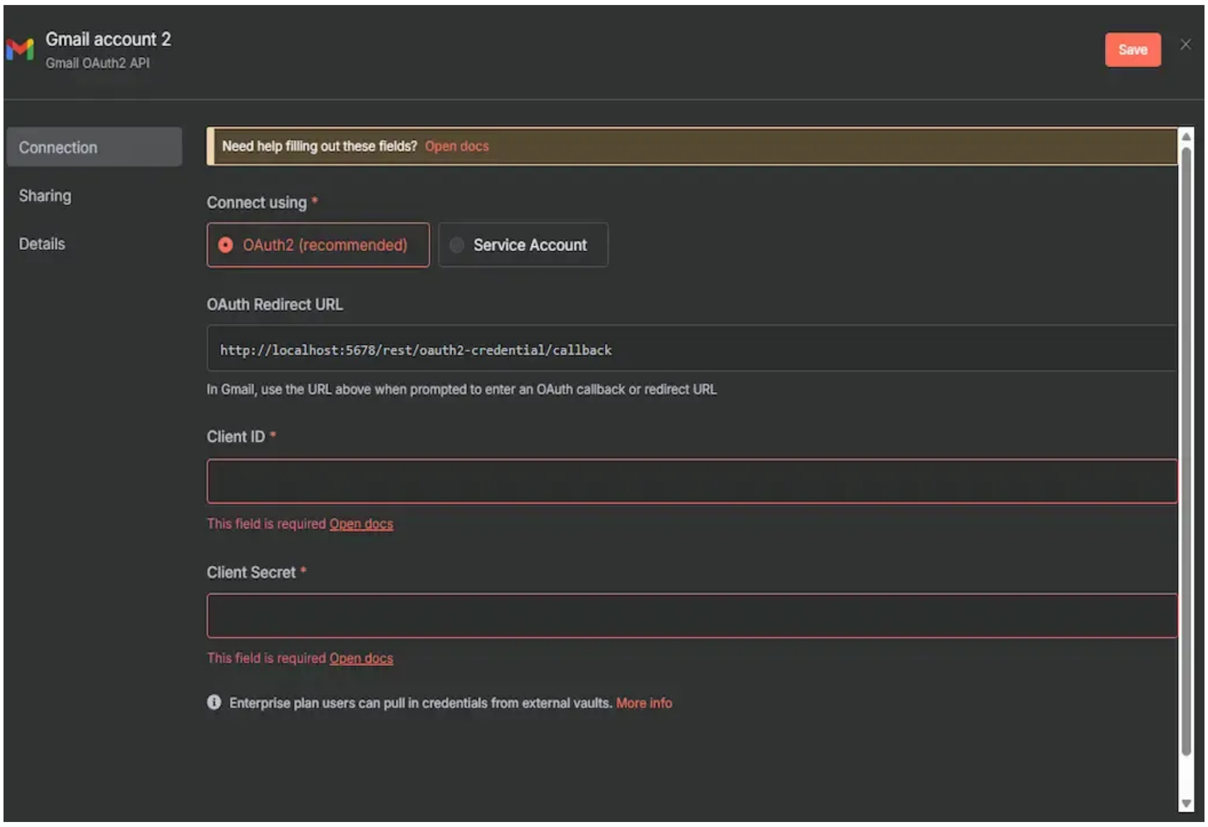
Gmail trigger

Step 6: Select **Create new credential** under the **Credentials to connect with**.



Create new credential

Step 7: Enter the Client ID and Client Secret that was extracted from google cloud console.



Need help filling out these fields? Open docs

Connect using *

[✓ OAuth2 (recommended)] Service Account

OAuth Redirect URL

http://localhost:5678/rest/oauth2-credential/callback

In Gmail, use the URL above when prompted to enter an OAuth callback or redirect URL

Client ID *

This field is required Open docs

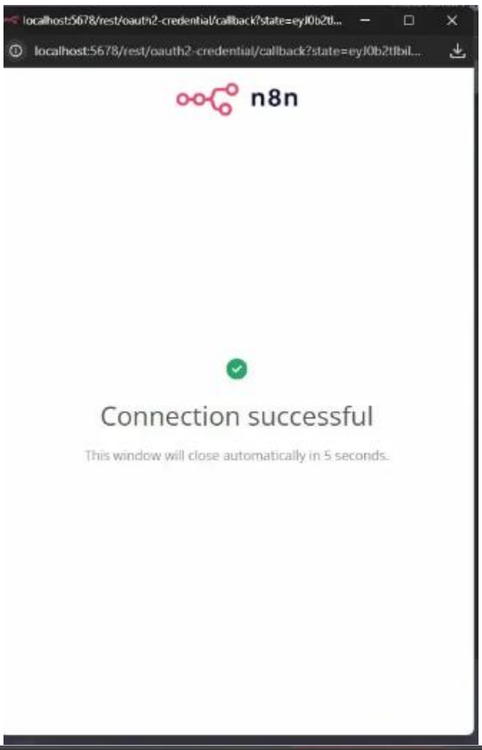
Client Secret *

This field is required Open docs

Enterprise plan users can pull in credentials from external vaults. More info

Data

Step 8: If entered details are correct and the login was successful, a pop-up will notify that account connected and you can also verify by executing the Gmail trigger.

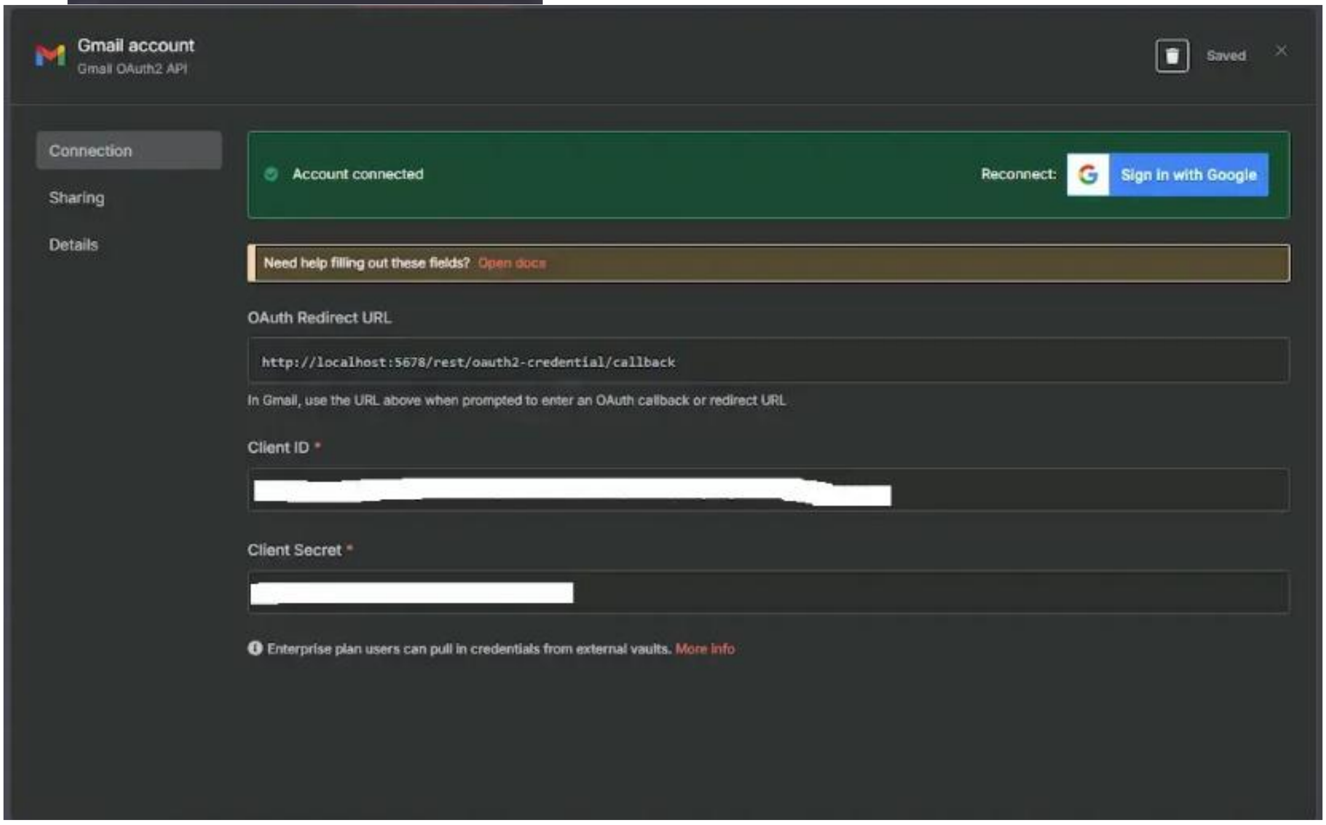


localhost:5678/rest/oauth2-credential/callback?state=eyJ0b2tl...

localhost:5678/rest/oauth2-credential/callback?state=eyJ0b2tlbiI...

Connection successful

This window will close automatically in 5 seconds.



Gmail account
Gmail OAuth2 API

Account connected

Reconnect: Sign in with Google

Need help filling out these fields? Open docs

OAuth Redirect URL

```
http://localhost:5678/rest/oauth2-credential/callback
```

In Gmail, use the URL above when prompted to enter an OAuth callback or redirect URL

Client ID *

[redacted]

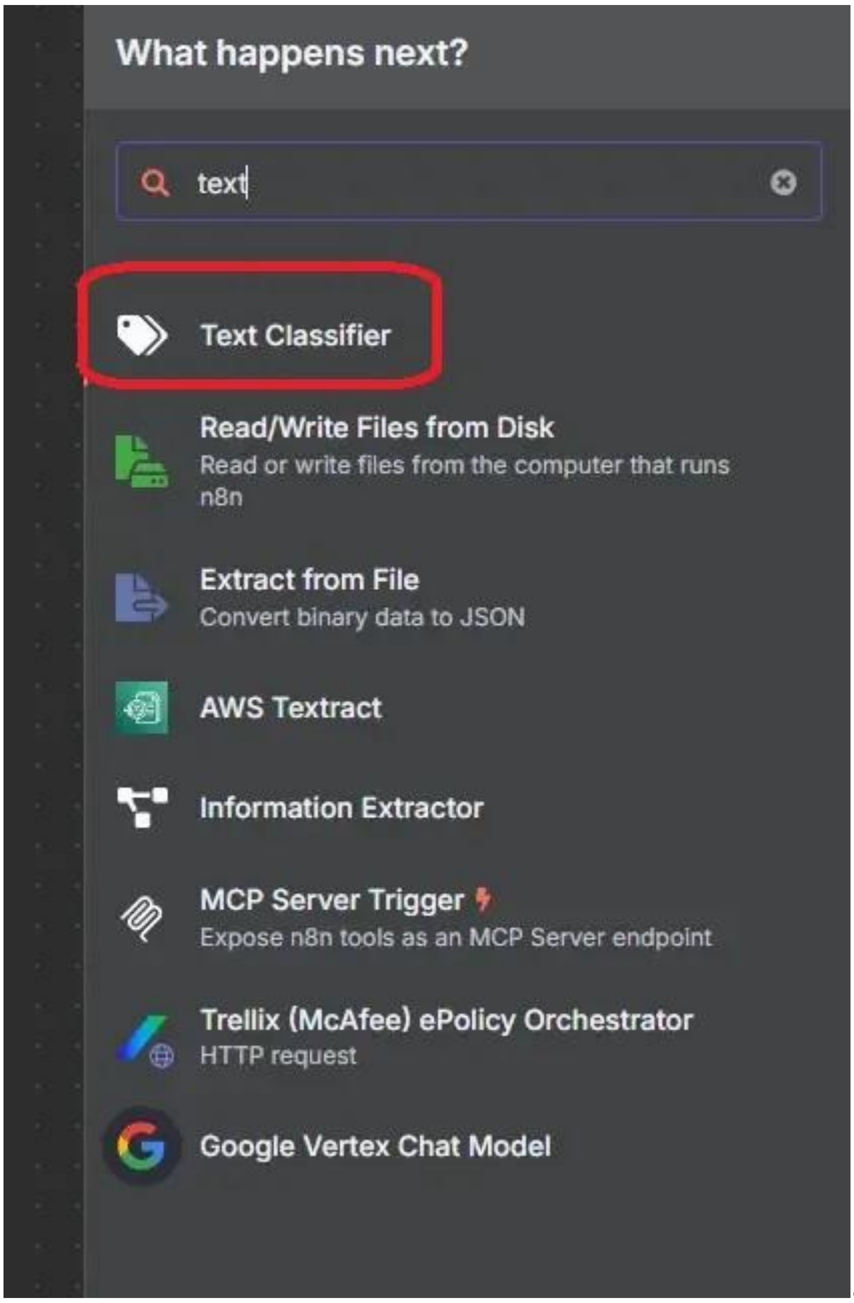
Client Secret *

[redacted]

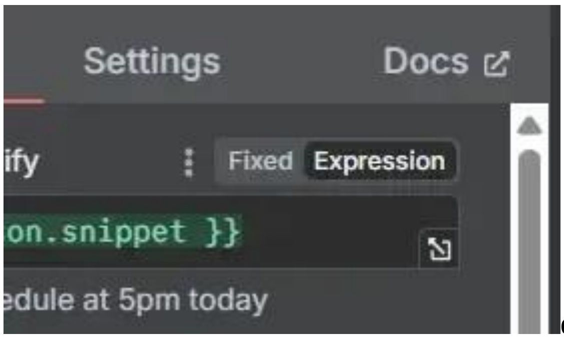
Enterprise plan users can pull in credentials from external vaults. More info

Creating Text Classifier Node

Step 1: Add another node, on the right top corner of the canvas and search for text classifier.

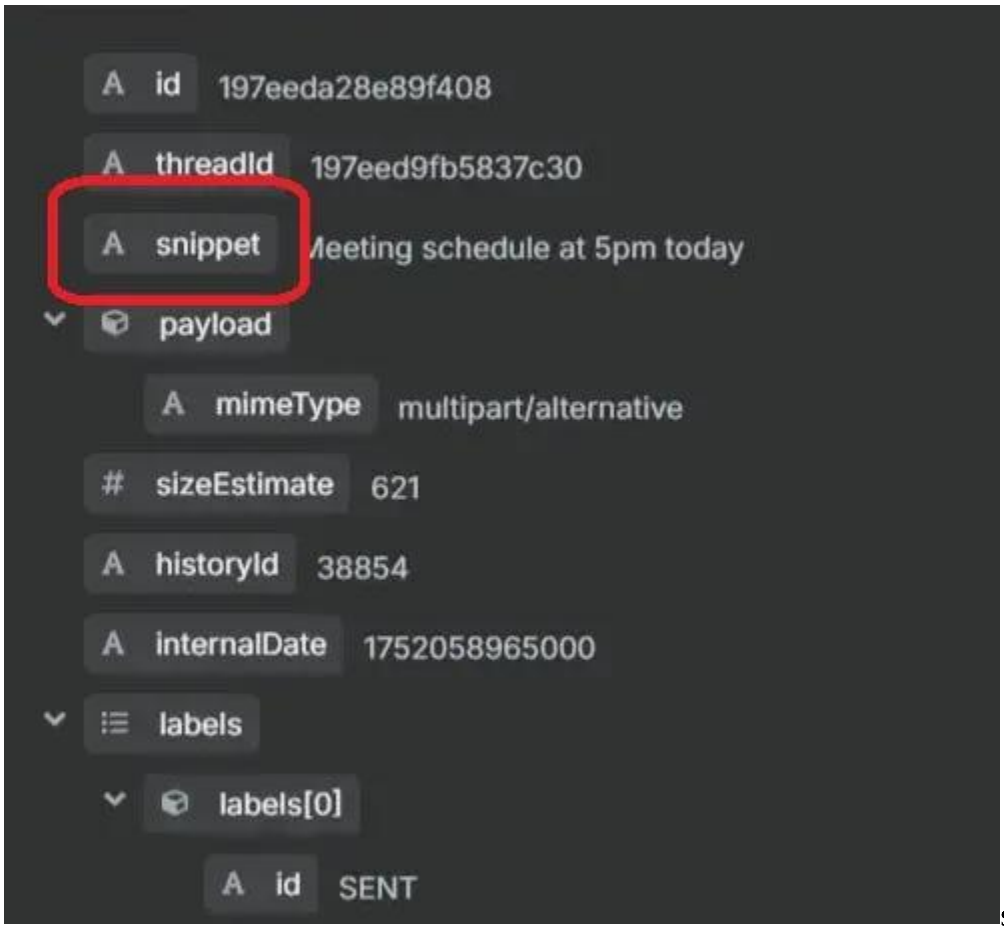

What happens next?
text
Text Classifier
Read/Write Files from Disk
Read or write files from the computer that runs n8n
Extract from File
Convert binary data to JSON
AWS Textract
Information Extractor
MCP Server Trigger
Expose n8n tools as an MCP Server endpoint
Trellix (McAfee) ePolicy Orchestrator
HTTP request
Google Vertex Chat Model
another node

Step 2: Inside the text classifier node, select the expression option from the fixed and expression.



expression

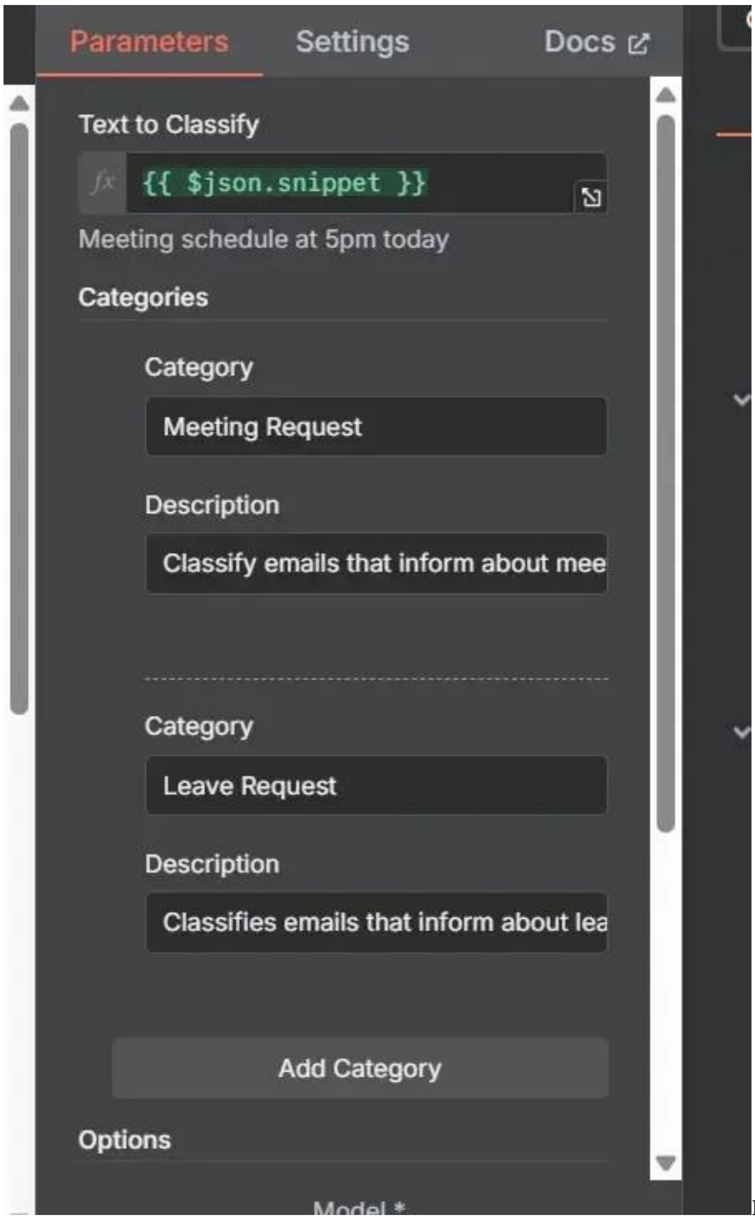
Step 3: Select the snippet from the schema of previous node and add it to the Text to classify box.



```
A id 197eeda28e89f408
A threadId 197eed9fb5837c30
A snippet Meeting schedule at 5pm today
payload
  A mimeType multipart/alternative
# sizeEstimate 621
A historyId 38854
A internalDate 1752058965000
labels
  labels[0]
    A id SENT
```

snippet

Step 4: Define the required categories such as Meeting Request and Leave Request.


Parameters Settings Docs

Text to Classify

```
{{ $json.snippet }}
```

Meeting schedule at 5pm today

Categories

Category: Meeting Request

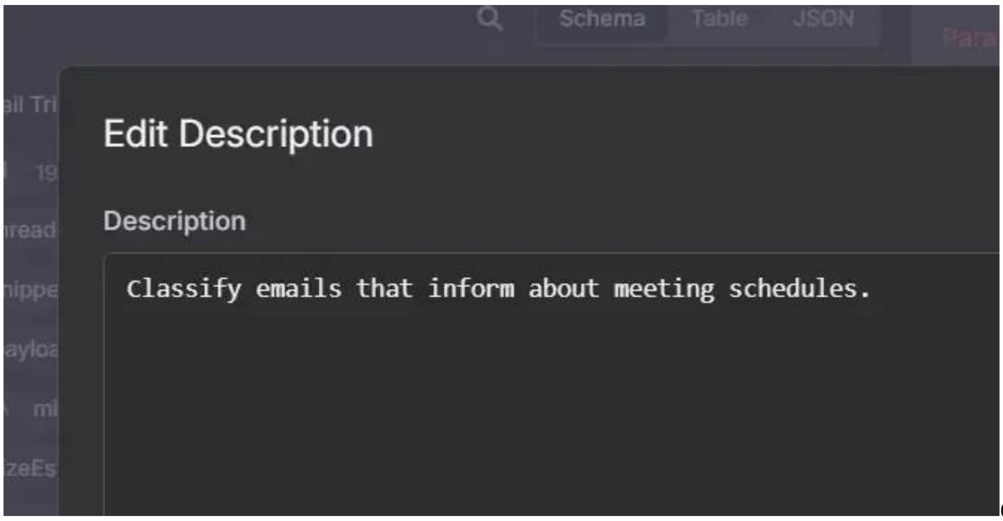
Description: Classify emails that inform about mee

Category: Leave Request

Description: Classifies emails that inform about lea

required categories

Step 5: Give the description of the categories.



Edit Description

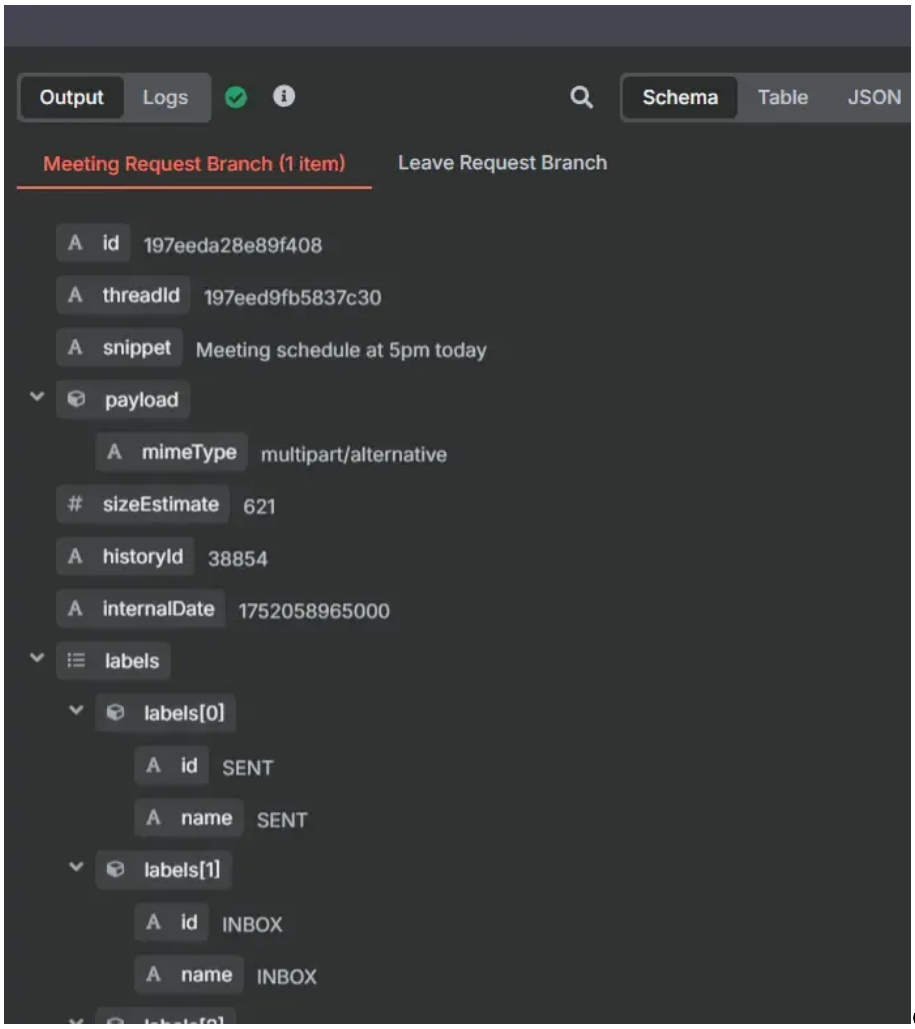
Description

```
Classify emails that inform about meeting schedules.
```

description

Step 6: Click on Add Option and select **System Prompt Template** from the drop box**.**

Step 7: Execute step and check for any issue.

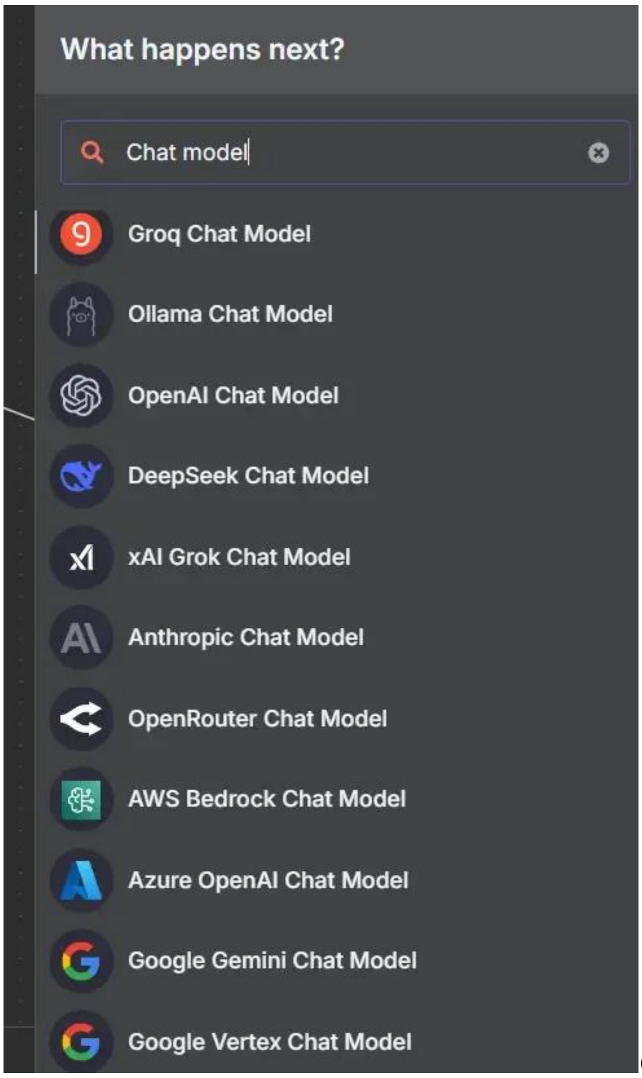


```
A id 197eeda28e89f408
A threadId 197eed9fb5837c30
A snippet Meeting schedule at 5pm today
payload
    A mimeType multipart/alternative
# sizeEstimate 621
A historyId 38854
A internalDate 1752058965000
labels
    labels[0]
        A id SENT
        A name SENT
    labels[1]
        A id INBOX
        A name INBOX
```

check for any issue

Creating Chat Model Node

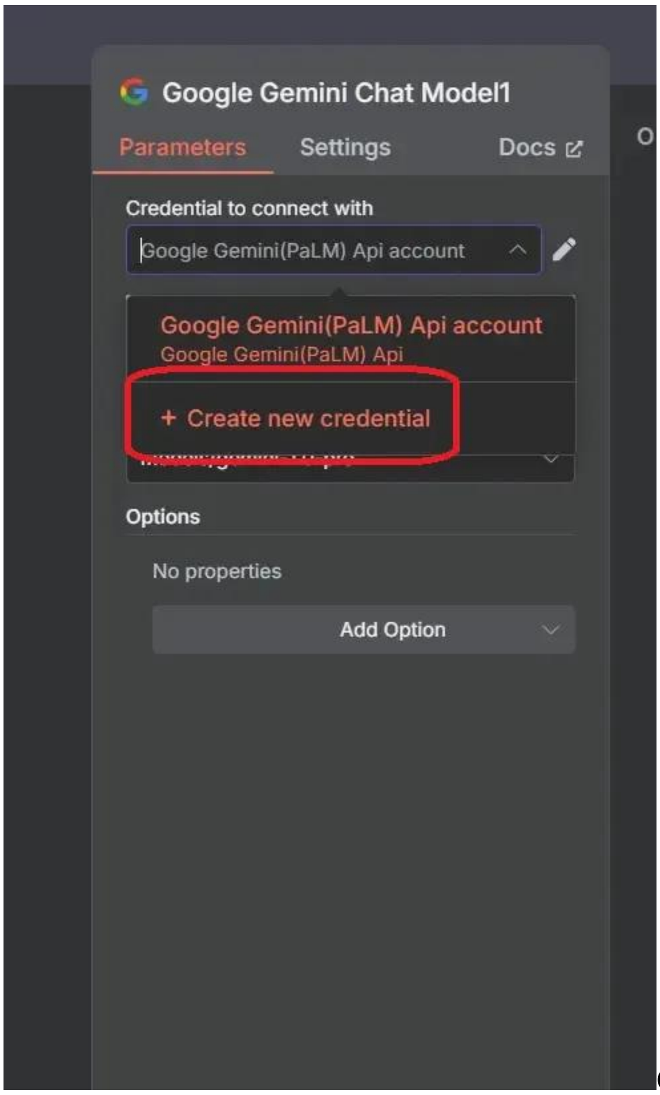
Step 1: From the top right corner of canvas, click on add node option and search for Chat model.



Chat model

Step 2: Choose the model that we want to use in our workflow, here we are going to use Google Gemini Chat Model.

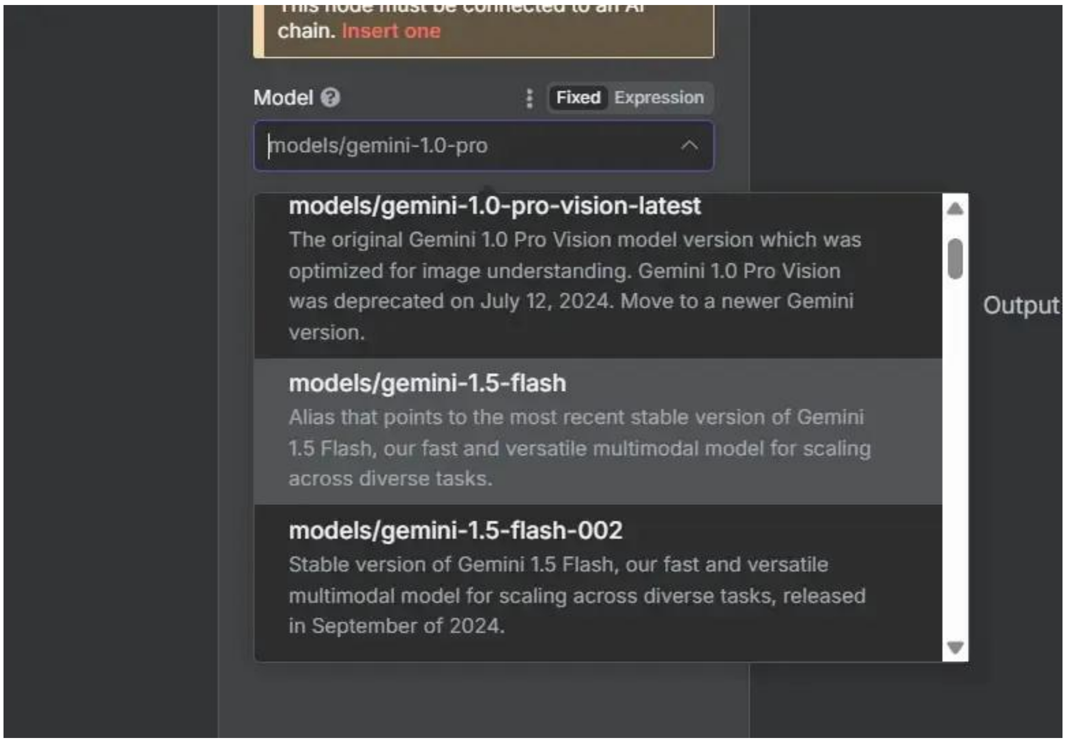
Step 3: Click on the Create New Credential.


Google Gemini Chat Model1
Parameters
Settings
Docs
Credential to connect with
Google Gemini(PaLM) Api account
Google Gemini(PaLM) Api account
Google Gemini(PaLM) Api
+ Create new credential
Options
No properties
Add Option

Create New Credential

Step 4: Enter the API Key extracted from the Google Gemini API key website.

Step 5: Select the model which we want to use in the workflow. Here we are using google gemini 1.5 flash.


This node must be connected to an AI chain. Insert one

Model — Fixed | Expression

models/gemini-1.0-pro

models/gemini-1.0-pro-vision-latest
The original Gemini 1.0 Pro Vision model version which was optimized for image understanding. Gemini 1.0 Pro Vision was deprecated on July 12, 2024. Move to a newer Gemini version.

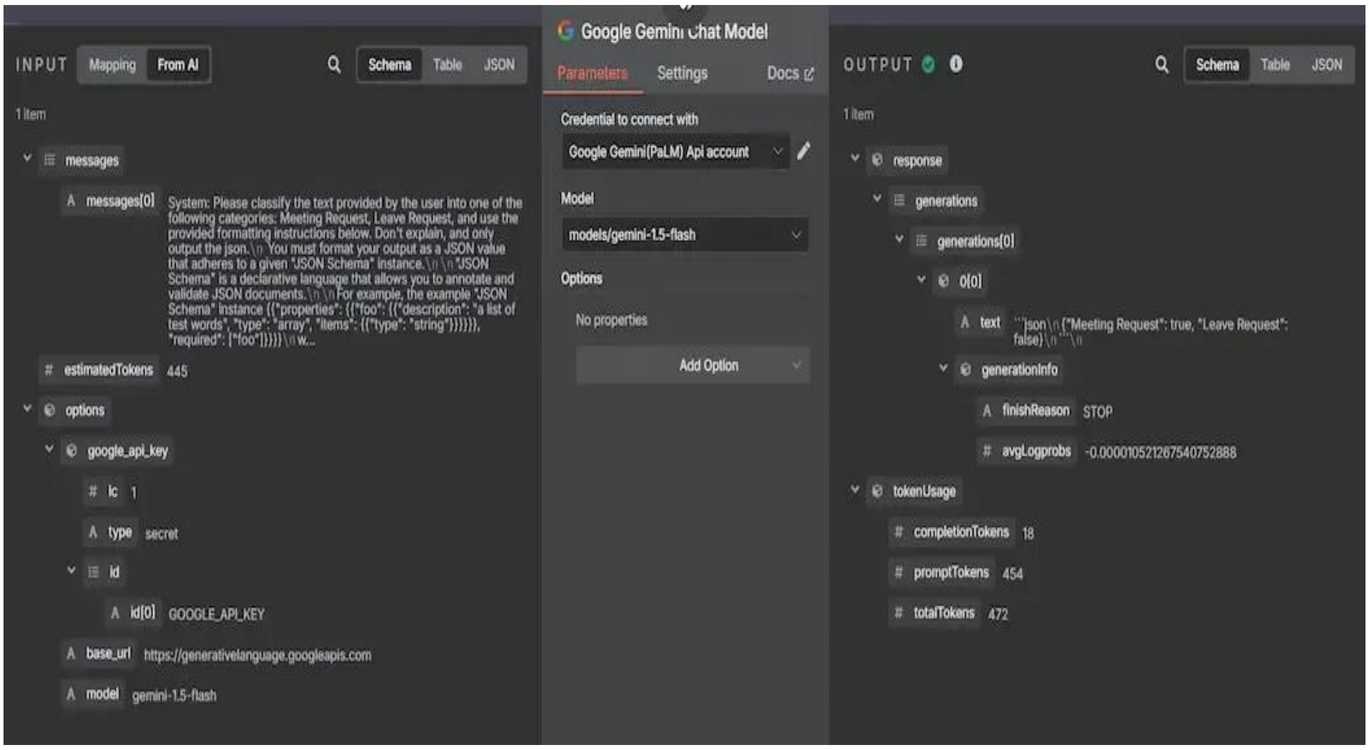
models/gemini-1.5-flash
Alias that points to the most recent stable version of Gemini 1.5 Flash, our fast and versatile multimodal model for scaling across diverse tasks.

models/gemini-1.5-flash-002
Stable version of Gemini 1.5 Flash, our fast and versatile multimodal model for scaling across diverse tasks, released in September of 2024.

Output

Select the model

Step 6: If the details are correct then it will run successfully.


```
INPUT   Mapping | From AI                  Schema | Table | JSON
1 item
messages
  messages[0]  System: Please classify the text provided by the user into one of the following categories: Meeting Request, Leave Request, and use the provided formatting instructions below. Don't explain, and only output the json.\n You must format your output as a JSON value that adheres to a given "JSON Schema" instance.\n\n "JSON Schema" is a declarative language that allows you to annotate and validate JSON documents.\n\n For example, the example "JSON Schema" instance {{"properties": {{"foo": {{"description": "a list of test words", "type": "array", "items": {{"type": "string"}}}}}}, "required": ["foo"]}}}}\n w...
estimatedTokens  445
options
  google_api_key
    lc  1
    type  secret
    id
      id[0]  GOOGLE_API_KEY
  base_url  https://generativelanguage.googleapis.com
  model  gemini-1.5-flash
```

```
Google Gemini Chat Model
Parameters   Settings   Docs
Credential to connect with
Google Gemini(PaLM) Api account
Model
models/gemini-1.5-flash
Options
No properties
Add Option
```

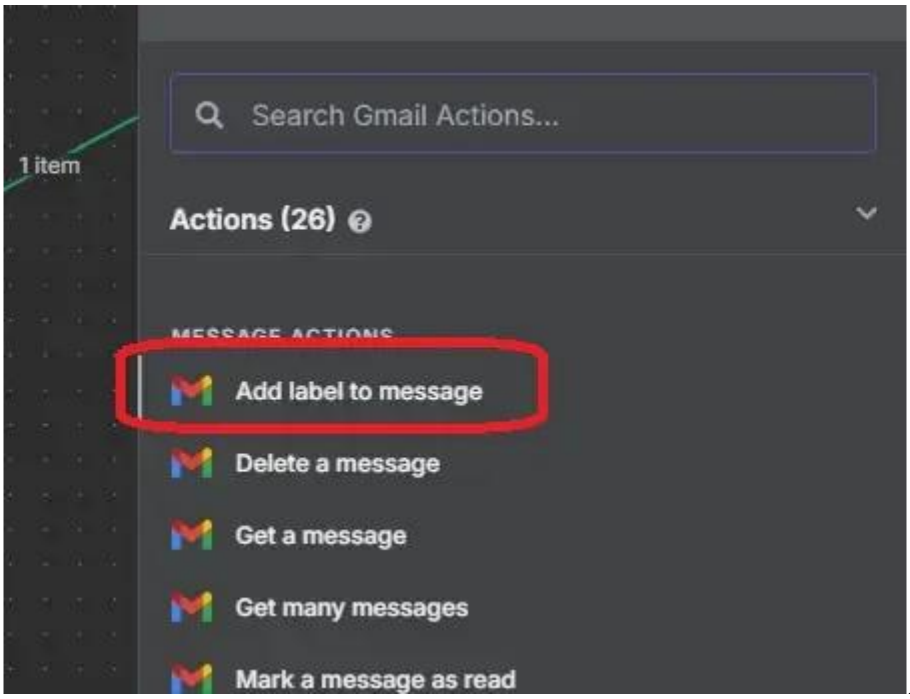
```
OUTPUT                                     Schema | Table | JSON
1 item
response
  generations
    generations[0]
      0[0]
        text  ```json\n{"Meeting Request": true, "Leave Request": false}\n```\n
        generationInfo
          finishReason  STOP
          avgLogprobs  -0.000010521267540752888
tokenUsage
  completionTokens  18
  promptTokens  454
  totalTokens  472
```

checking

Creating Meeting Request and Leave Request Nodes

Step 1: From the add new node option, search for Gmail and click on it.

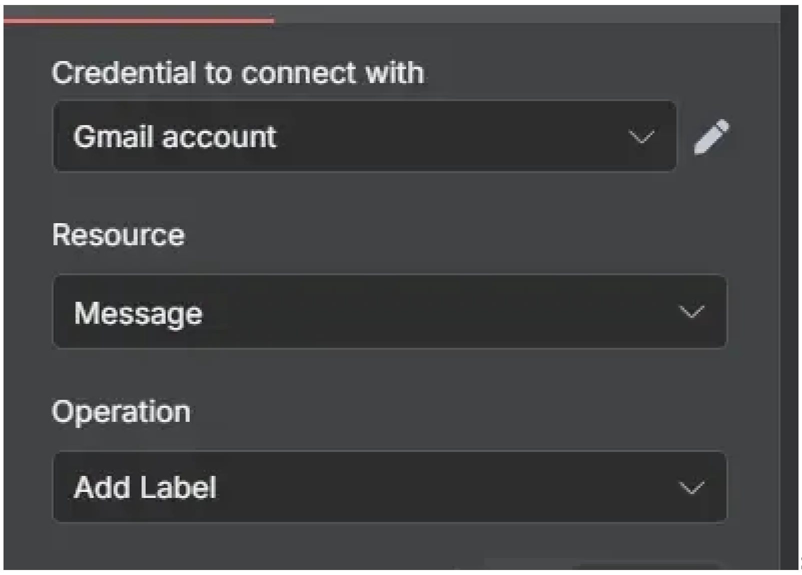
Step 2: From the list of actions, select Add label to message.



Add label to message

Step 3: Attach the same Gmail account that was used in the Gmail trigger Credentials.

Step 4: Select Message in the Resource option. And select Add Label operation.

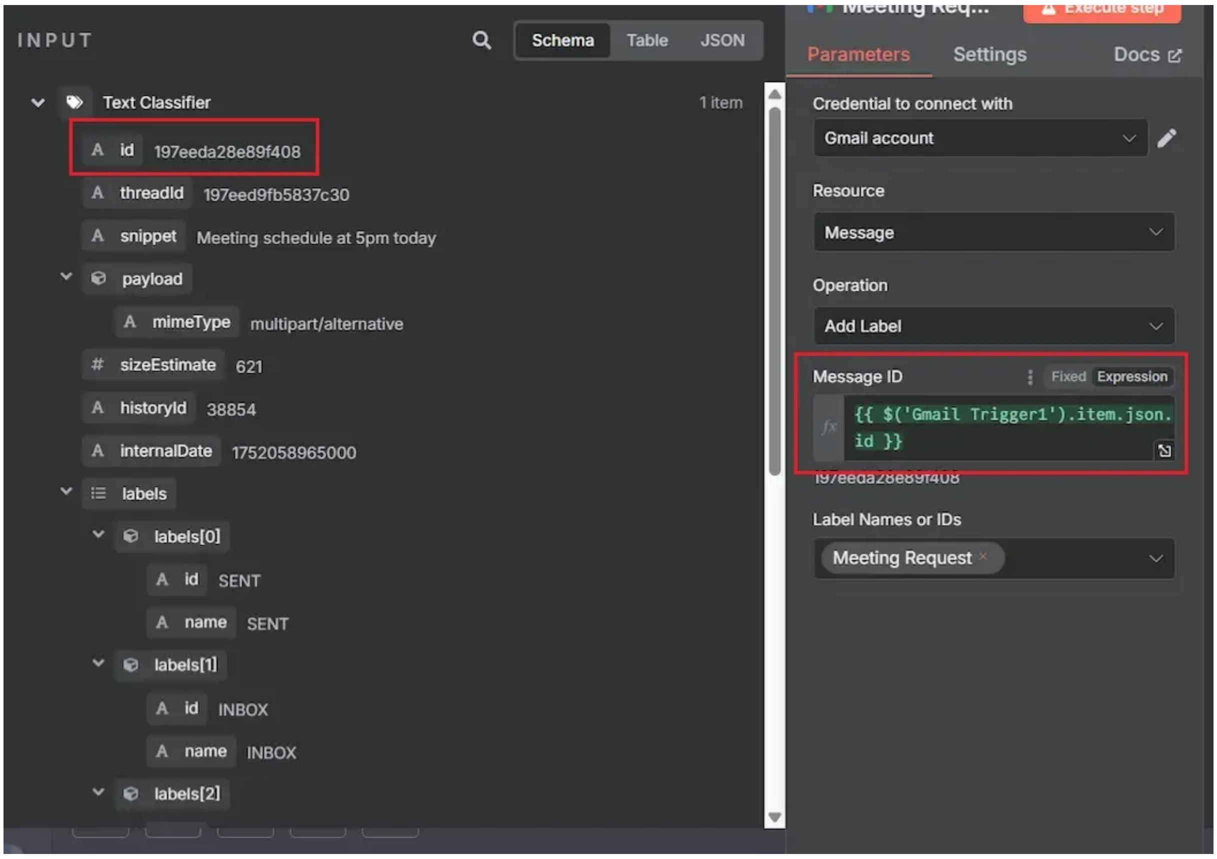

Credential to connect with: Gmail account

Resource: Message

Operation: Add Label

select Add Label operation

Step 5: In the message ID Dialog box, paste ID from the Input schema.


INPUT

Schema | Table | JSON

Text Classifier — 1 item

```
A id 197eeda28e89f408
A threadId 197eed9fb5837c30
A snippet Meeting schedule at 5pm today
payload
    A mimeType multipart/alternative
# sizeEstimate 621
A historyId 38854
A internalDate 1752058965000
labels
    labels[0]
        A id SENT
        A name SENT
    labels[1]
        A id INBOX
        A name INBOX
    labels[2]
```

Meeting Req... — Execute step

Parameters | Settings | Docs

Credential to connect with: Gmail account

Resource: Message

Operation: Add Label

Message ID (Fixed | Expression):

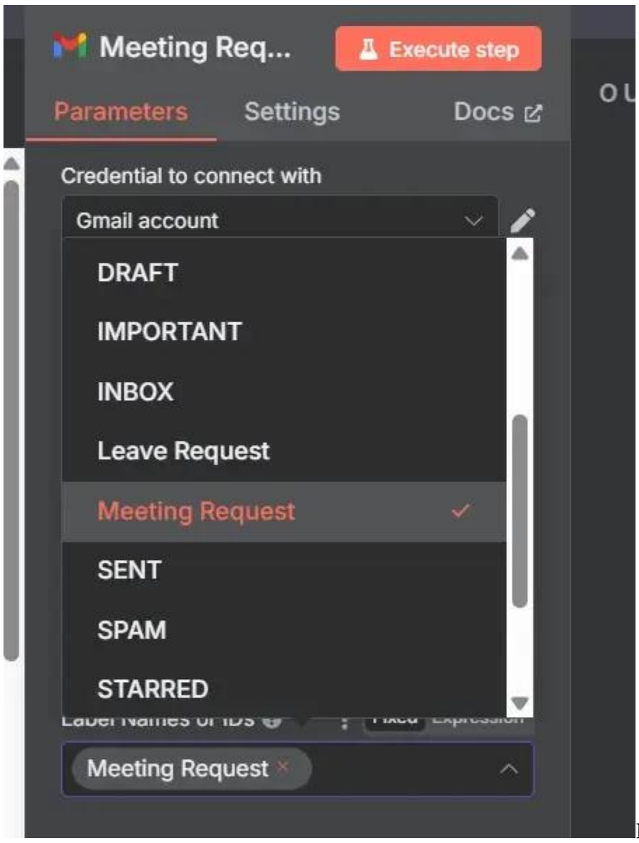
```
{{ $('Gmail Trigger1').item.json.id }}
```

197eeda28e89f408

Label Names or IDs: Meeting Request

paste ID

Step 6: Add the required label from the drop box.


Meeting Req... — Execute step

Parameters | Settings | Docs

Credential to connect with: Gmail account

Label Names or IDs: Meeting Request

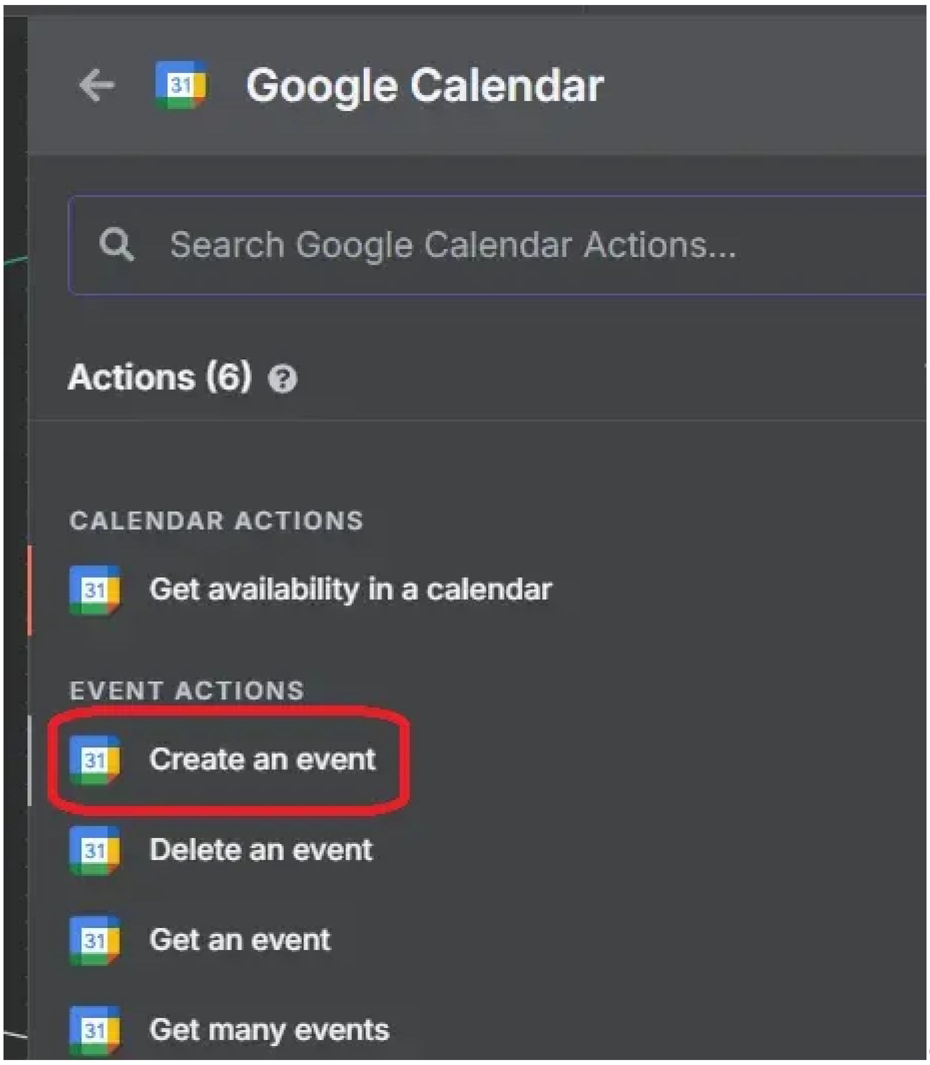
required label

Step 7: Execute the step to check for any issues, if the workflow works fine the output will be shown.

Creating the Google Calendar Node

Step 1: From the add new node option, search for Google Calendar.

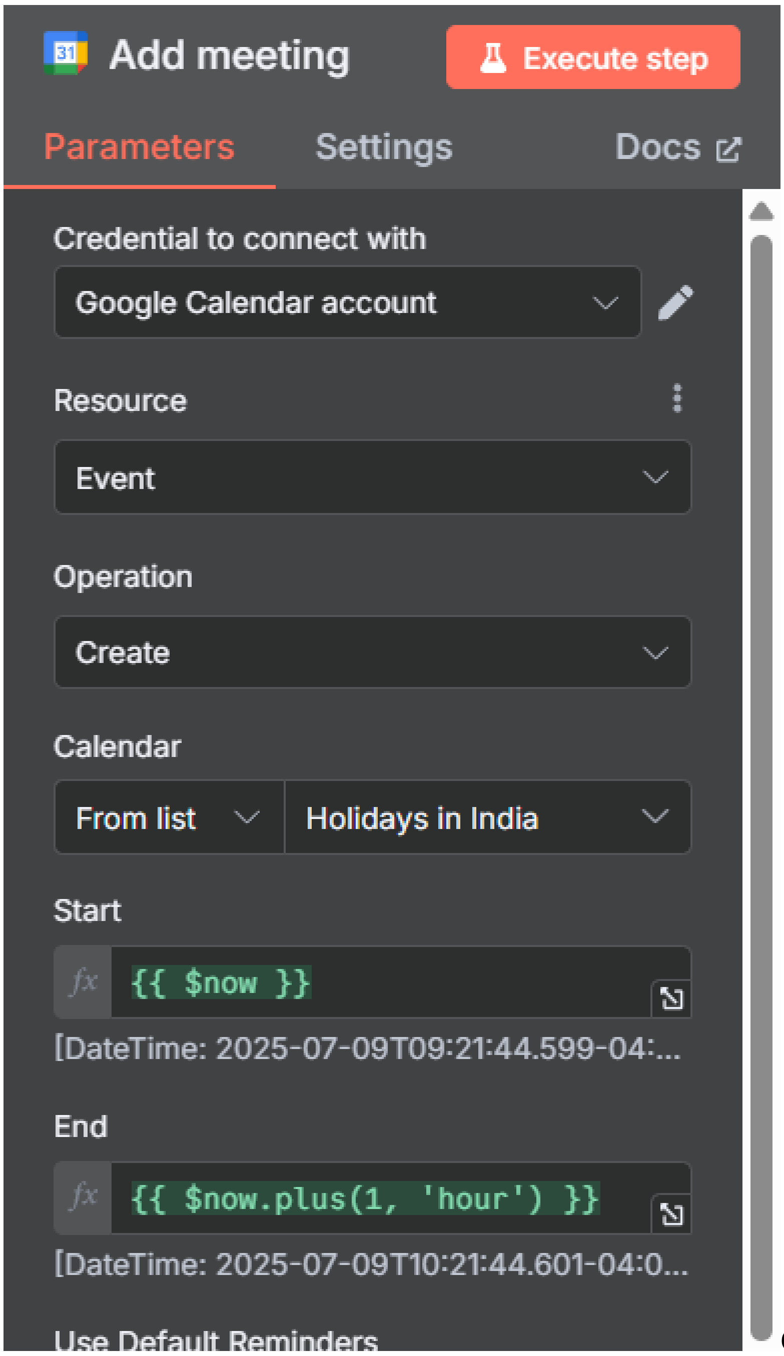
Step 2: Search for Create an event under the Event Actions.



Create an event

Step 3: Attach the same Gmail account that was used in the Gmail trigger Credentials.

Step 4: Select the Create operation, select the Calendar and define the Start and End.


Add meeting

Execute step

Parameters Settings Docs

Credential to connect with

Google Calendar account

Resource

Event

Operation

Create

Calendar

From list | Holidays in India

Start

```
{{ $now }}
```

[DateTime: 2025-07-09T09:21:44.599-04:...

End

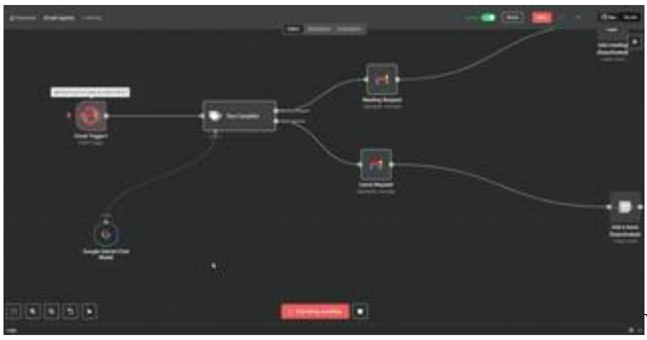
```
{{ $now.plus(1, 'hour') }}
```

[DateTime: 2025-07-09T10:21:44.601-04:0...

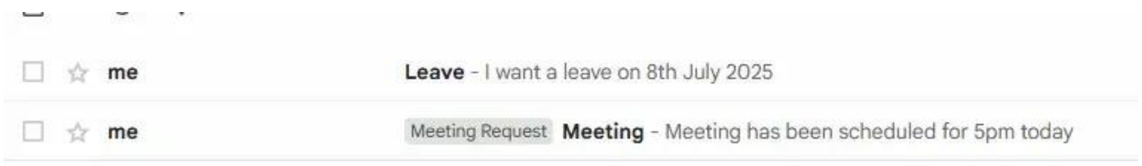
Use Default Reminders

Create operation

Workflow Execution


Workflow Execution

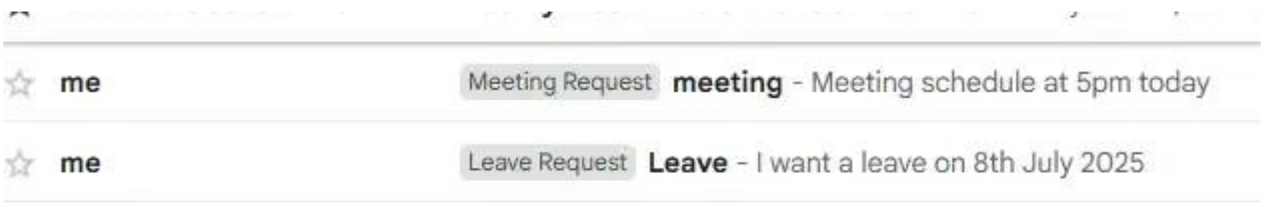
Before execution.


me Leave - I want a leave on 8th July 2025

me Meeting Request Meeting - Meeting has been scheduled for 5pm today

Before execution

After execution.

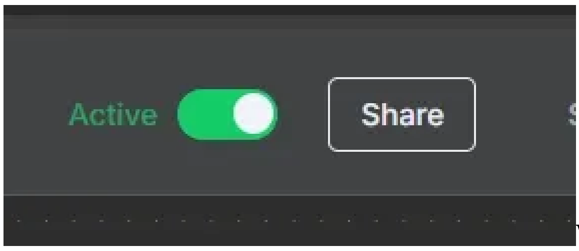

me Meeting Request meeting - Meeting schedule at 5pm today

me Leave Request Leave - I want a leave on 8th July 2025

After execution

Workflow Activation

After the workflow is created, we can activate it and now it will automatically perform its task on a fixed interval of time.



Workflow Activation