



AppScript Generative AI (Gemini API)

Sr. Full-Stack Engineer

Welly Zhang



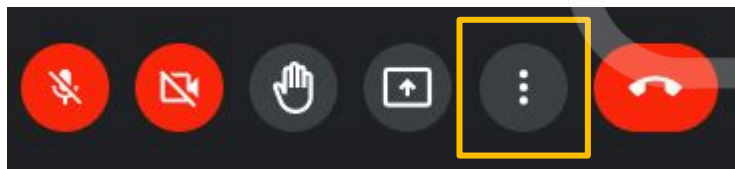
CloudMile

在會議中 有任何疑問時

1. 請點擊 Meet 右上角的即時「問與答」功能提出您的問題！
2. 大家可以點擊互動按讚，我們將會優先回答按讚數較多的問題！



為了確保 會議品質



變更版面配置

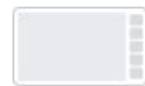
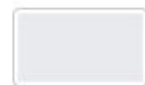
系統會記住你選取的版面配置，以供日後的會議使用

自動

圖塊

聚光燈

側欄



並排顯示

無法變更這個版面配置的圖塊數量



Welly Zhang

Data Application & ML Team Lead

Sr. Full-Stack Engineer

Solution Architect

擁有 6 年以上全端工程師、1 年 Solution Architect 以及 2 年 Tech Team Lead 經驗。

現任職 CloudMile 萬里雲開發團隊及機器學習團隊主管，幫助多家大型企業如金融業、製造業、半導體業、零售業用戶在 CloudMile Google 雲端生態圈建立雲端搬遷之解決方案、數據中台建置解決方案以及 GWS 相關服務應用 ex. AppSheet

- # Google Workspace for Developer
- # Google Apps Script
- # Google AppSheet
- # GCP Data Platform Solution
- # Full-Stack Development
- # Devops Solution



AGENDA

01



Gemini Overview

02



Prompt Engineering

03



使用 AppScript 呼叫 Gemini API

04



利用 RAG 技術, 將 Gmail 內容提供給 Gemini 做信件整理

01

Gemini Overview

CloudMile

Gemini Introduction

◆ Gemini for  





Vertex AI

AI Solutions

Contact Center | Document | Discovery | Risk | Healthcare

Agent Builder (Search & Conversation)

AI Platform

Extension | Connectors | Grounding

Prompt | Serve | Tune | Distill | Eval

Notebooks | Training | Feature Store | Pipelines | Monitoring

Notebooks | Training | Feature Store | Pipelines | Monitoring

Model Garden

Google | OSS | Partner Models

Google Cloud Infrastructure (GPU / TPU) | Data Cloud



商業用戶



開發者



AI專家/ 研究員

Gemini Model

Model Garden on Vertex AI

Gemini foundation models	Gemini 1.0 Pro	Gemini 1.5 Pro	Gemini 1.5 Flash				
Google foundation models	PaLM 2	Imagen 2	Chirp	Codey	Embeddings		
Google task specific models	Speech-to-Text Enterprise Document OCR	Text-to-Speech	Occupancy analytics	Cloud Natural Language API	Vision API	Translation API	Cloud Video Intelligence API
Google domain specific models	MedLM Life Science and Healthcare	Sec-PaLM Cybersecurity					
Partner & Open Ecosystem	Llama 3 Code Llama	Falcon	AI	Claude 2 Pre-announce	MISTRAL AI	Gemma	

- Google、開源和第三方基礎模型的**選擇和靈活性**
- **多模態**可匹配每個 Use case
- **多種模型大小**可滿足成本和功效需求
- 針對專門行業的**特定領域模型**
- 企業級**安全、保障和責任**
- 透過**完全整合的平台**縮短價值實現時間

Google AI for Developer

Google AI for Developers Gemini API Gemma Google AI Edge Tools Community

Search English Sign in

Docs API Reference Cookbook Google AI Studio Prompt gallery Pricing

Build with the Gemini API

Easily integrate Google's largest and most capable AI model to your apps

[Get API key in Google AI Studio](#) [Read API docs](#)

[Try Google Cloud's enterprise-ready AI >](#)

Link: <https://ai.google.dev/gemini-api>

Get API Key for Gemini API

Gemini 1.5 Pro 2M context window, code execution capabilities, and Gemma 2 are now available. [Learn more.](#) Got it

Google AI Studio

- Get API key
- Create new prompt
- New tuned model
- My library
- Allow Drive access

- Getting started
- Documentation
- Prompt gallery
- Gemini cookbook
- Discourse forum
- Build with Vertex AI on Google Cloud

- Settings

Get API key

API keys

You can create a new project if you don't have one already or add API keys to an existing project. All projects are subject to the [Google Cloud Platform Terms of Service](#), which you agree to when creating a new project, while use of the Gemini API and Google AI Studio is subject to the [Gemini API Terms of Service](#).

Use your API keys securely. Do not share them or embed them in code the public can view.

If you use Gemini API from a project that has billing enabled, your use will be subject to [pay-as-you-go pricing](#).

[Create API key](#)

Your API keys are listed below. You can also [view and manage your project and API keys in Google Cloud](#).

Project number	Project ID	API key	Created	Plan
...6982	Generative Language Client	...RS_k	Dec 20, 2023	Free of charge Set up Billing View usage data

Quickly test the API by running a cURL command

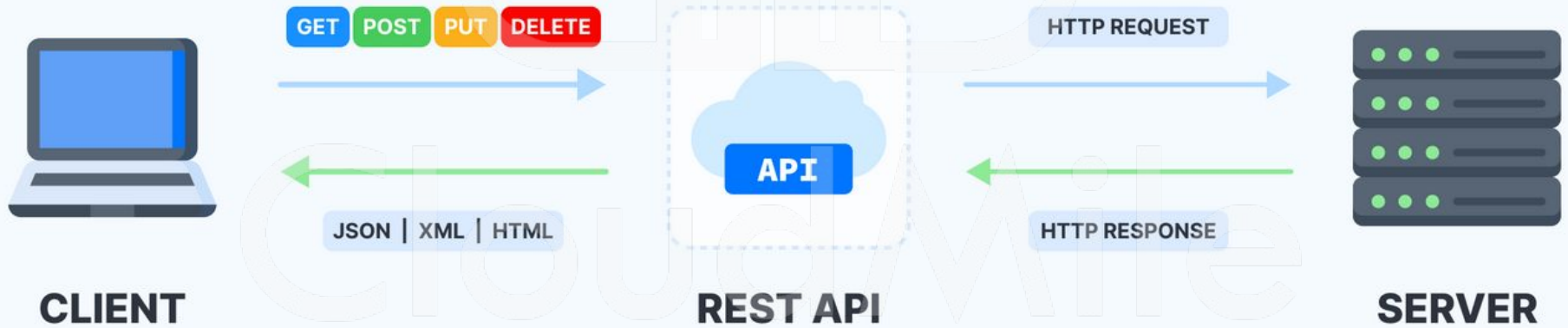
```
curl \
-H 'Content-Type: application/json' \
-d '{"contents":[{"parts":[{"text":"Explain how AI works"}]}]}' \
-X POST 'https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5-flash-latest:generateContent?key=YOUR_API_KEY'
```

[Use code with caution.](#)

[API quickstart guide](#)

RESTful API

REST API Model



HTTP methods in RESTful API development

HTTP Methods	Path	Action
GET	/products/{sku}	Read
POST	/products	Create
PUT	/products/{sku}	Update
PATCH	/products/{sku}	Partial Update
DELETE	/products/{sku}	Delete

Try: Testing for your first Gemini API call

```
-zsh
Last login: Tue Jul 23 15:18:37 on ttys001
welly@zhangjiaweis-MacBook-Pro ~ % curl \
-H 'Content-Type: application/json' \
-d '{"contents":[{"parts":[{"text":"Explain how AI works"}]}]}' \
-X POST 'https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5-flash-latest:generateContent?key=YOUR_API_KEY'
{"candidates": [
  {
    "content": {
      "parts": [
        {
          "text": "## Understanding the Magic of AI: A Simple Explanation\n\nArtificial intelligence (AI) is essentially making computers think and learn like humans, but without the biological limitations. It's a broad field, encompassing many different approaches, but at its core, it relies on three key components:\n\n**1. Data:** AI systems need vast amounts of data to learn. This data can be anything from images and text to sensor readings and financial data. The more diverse and high-quality the data, the better the AI system will perform.\n\n**2. Algorithms:** These are the sets of instructions that tell the computer how to process the data and learn from it. Different algorithms are designed for different tasks, like image recognition, natural language processing, or predicting future events.\n\n**3. Learning:** This is the core of AI. Through algorithms, AI systems learn patterns and relationships from the data, allowing them to make predictions, solve problems, and complete tasks. There are two main types of learning:\n\n**Supervised learning:** The system is trained on labeled data, where the correct answers are provided. For example, showing the AI pictures of cats and labeling them \"cat\" helps it learn to recognize cats in future images.\n\n**Unsupervised learning:** The system learns from unlabeled data, uncovering hidden patterns and structures. For example, grouping similar customers together based on their purchasing behavior.\n\n**Think of AI as a student:**\n\n**Data:** The textbooks, lectures, and real-world experiences that a student learns from.\n\n**Algorithms:** The study methods and learning strategies used to absorb the knowledge.\n\n**Learning:** The process of acquiring knowledge and applying it to solve problems.\n\n**How AI is used today:**\n\nAI is revolutionizing many industries, from healthcare to finance to entertainment. Some common examples include:\n\n**Self-driving cars:** Using sensors and algorithms to navigate roads autonomously.\n\n**Facial recognition:** Identifying individuals based on their facial features.\n\n**Virtual assistants:** Responding to voice commands and answering questions.\n\n**Personalized recommendations:** Suggesting products and services based on user preferences.\n\n**Medical diagnosis:** Assisting doctors in identifying and diagnosing diseases.\n\n**The future of AI:**\n\nAI is continuously evolving, with new algorithms and applications being developed every day. As computing power increases and data availability expands, AI will become even more powerful and transformative, impacting all aspects of our lives.\n\n**Important note:** AI is a complex field with many nuances. This explanation provides a high-level overview, and there's much more to learn about the different types of AI, their capabilities, and their ethical implications.\n\n"}
        ]
      }
    },
    "role": "model"
  },
  "finishReason": "STOP",
  "index": 0,
  "safetyRatings": [
    {
      "category": "HARM_CATEGORY_SEXUALLY_EXPLICIT",
      "probability": "NEGLIGIBLE"
    },
    {
      "category": "HARM_CATEGORY_HATE_SPEECH",
      "probability": "NEGLIGIBLE"
    },
    {
      "category": "HARM_CATEGORY_HARASSMENT",
      "probability": "NEGLIGIBLE"
    }
  ]
}]
```

```
curl \
-H 'Content-Type:
application/json' \
-d
'{"contents":[{"parts":[{"t
ext":"Explain how AI
works"}]}]}' \
-X POST
'https://generativelangu
age.googleapis.com/v1b
eta/models/gemini-1.5-
flash-latest:generateCo
ntent?key=YOUR_API_K
EY'
```

02

Prompt Engineering

CloudMile

何謂 Prompt Engineering?

提示工程 (Prompt Engineering) 是引導生成式人工智慧 (生成式 AI) 解決方案以產生所需輸出的程序。儘管生成式 AI 設法模仿人類，但它仍需要詳細的指示來建立高品質和相關的輸出。在提示工程中，您可以選擇最合適的格式、片語、單詞和符號，以引導 AI 更有意義地與使用者互動。提示工程師使用創意加試錯法來建立輸入文字的集合，因此應用程式的生成式 AI 能夠如預期運作。

Six strategies for getting better results

- Write clear instructions (精準用字, 講重點, ex. 限制字數、確認回應範圍)
- Provide reference text (給予範例)
- Split complex tasks into simpler subtasks (將問題切割)
- Give the model time to "think" (Chain of Thought, 給予回答邏輯範例)
- Use external tools (Retrieval-Augmented Generation - RAG)
- Test changes systematically (給予範例並自動測試回覆結果是否合理)

Source: Open AI (<https://platform.openai.com/docs/guides/prompt-engineering>)

Task: 熟悉以下 Prompt Engineering 技巧

重點練習

清楚問題

台灣有多少城市，首都在哪？氣候如何？

角色扮演

你是一位台灣資深地理學家，請問台灣有多少城市，首都在哪？氣候如何？

Zero/One Shot

南韓的首都是首爾，但他並不是南韓最大的城市，詢問內容：台灣的首都？

Chain of Thought

台灣的首都是哪裡？請詳細解釋首都的特徵和其重要性。核心問題、基本知識、首都的特徵、首都的重要性、總結回答

Prompt Engineering 範例

幫我分析的以下的句子的情緒，最後再分類成正面情緒或負面情緒

句子: 這間燒臘便當很油。

推論:「油膩」通常被認為是一個不好的特點，可能會影響消費者的口感和健康。

情緒分類: 負面情緒

句子: 我迫不及待要學習新的 Google Cloud Platform 的知識了

推論: 這句話表達了一種積極主動的態度，並且顯示出對學習新知識的興奮和熱情。

情緒分類: 正面情緒

句子: <你要預測的句子>

Prompt Engineering 範例解析

instruction

in-context
learning

幫我分析的以下的句子的情緒，最後再分類成正面情緒或負面情緒

句子: 這間燒臘便當很油。

推論:「油膩」通常被認為是一個不好的特點，可能會影響消費者的口感和健康。

情緒分類: 負面情緒

chain-of-thought

句子: 我迫不及待要學習新的google cloud platform的知識了

推論: 這句話表達了一種積極主動的態度，並且顯示出對學習新知識的興奮和熱情。

情緒分類: 正面情緒

句子: <你要預測的句子>

03

使用 AppScript 呼叫 Gemini API

CloudMile

Let's go with Apps Script!

您需要先完成以下幾件事：

1. A Google Account (登入您的 Google 帳戶)
(Google Workspace accounts might require administrator approval)
2. A web browser with access to the internet (已連上網路的網頁頁面)
3. 開啟「Apps Script」頁面
<https://script.google.com/home>
4. 開啟「練習程式碼」文件
https://github.com/CloudMilePMO/CloudMile_NTU_Training/blob/main/20240809%20-%20GWS%20Apps%20Script%20for%20Gemini%20API

Apps Script 在開發前有哪些注意事項？

1. Apps Script IDE
2. Trigger
3. Deploy
4. Log
5. Project Setting
6. OAuth Setting

Apps Script demo_for_course1

Project Settings

General settings

Settings that pertain to the entire Apps Script project. Changes to these settings will not impact your existing deployment.

Time zone
(GMT+08:00) Taipei Standard Time - Taipei

Log uncaught exceptions to Cloud logs

Enable Chrome V8 runtime

Show "appscript.json" manifest file in editor

IDs

IDs are the unique identifiers of your Apps Script project.

Script ID 119LQ-aMLPgBHDcwmtQbvY6K0ks0yiPkgyiRr49F6bLOUDjvgjrcOSfJn
[Copy](#)

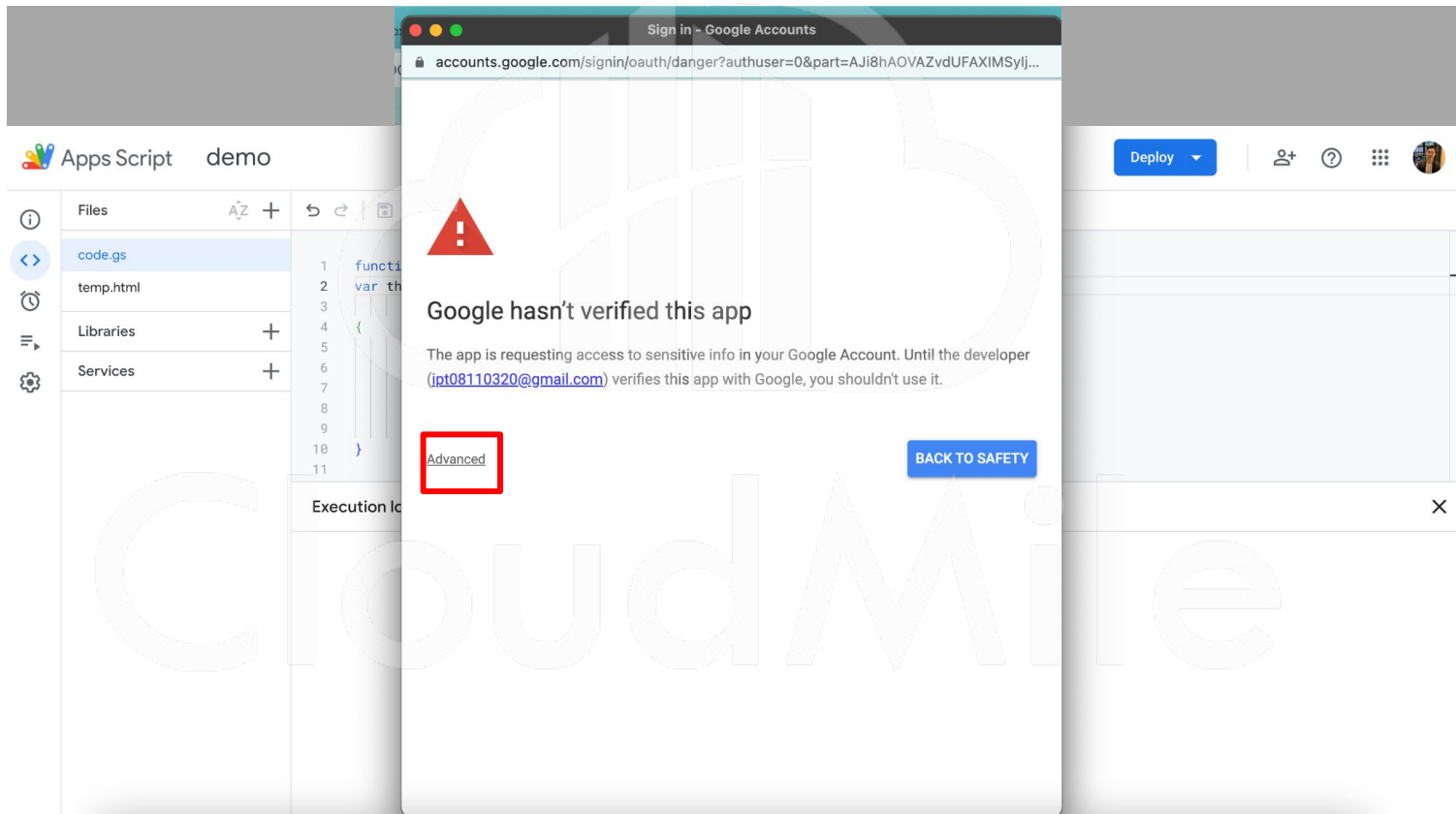
Google Cloud Platform (GCP) Project

Apps Script projects use Google Cloud Platform to manage authorizations, advanced services, and other details. To learn more, visit [Google Cloud Platform](#).

GCP Standard

Project Number 880025127837

執行時是否有出現這個畫面？



Properties

Comparison of property stores

The `PropertiesService` global object offers three methods, each of which returns a similar `Properties` object but with different access rights, as shown in the following table:

	Script Properties	User Properties	Document Properties
Method to access	<code>getScriptProperties()</code>	<code>getUserProperties()</code>	<code>getDocumentProperties()</code>
Data shared among	All users of a script, add-on, or web app	The current user of a script, add-on, or web app	All users of an add-on in the open document
Typically used for	App-wide configuration data, like the username and password for the developer's external database	User-specific settings, like metric or imperial units	Document-specific data, like the source URL for an embedded chart

Task 0: Let's try PropertiesService (Key-Value) code: 2.1

Apps Script NTU-04-Gemini

```
Files  AZ +  Run  Debug  generateContent  Execution log
appscript.json
Code.gs
Libraries +
Services +

1 function getGeminiApiKey() {
2   const scriptProperties = PropertiesService.getScriptProperties();
3   const geminiApiKey = scriptProperties.getProperty('GEMINI_API_KEY');
4   console.log(geminiApiKey);
5   return geminiApiKey;
6 }
7
```

Apps Script NTU-04-Gemini

- Log uncaught exceptions to Cloud logs
- Enable Chrome V8 runtime
- Show "appscript.json" manifest file in editor

IDs
IDs are the unique identifiers of your Apps Script project.

Script ID 1f23QyimmG_oxlbHf2qegfk6_o7UPQ4ZYOGKJA79JwAZQJn0TPv2UMS
[Copy](#)

Google Cloud Platform (GCP) Project

Apps Script projects use Google Cloud Platform to manage authorizations, advanced services, and other details. To learn more, visit [Google Cloud Platform](#).

GCP Default

[Change project](#)

Script Properties

Script properties provide a simple and powerful way of defining and exposing custom properties for a specific object instance. Learn more in [script properties documentation](#).

Property	Value
GEMINI_API_KEY	AlzaSyDgTJ-mwRvF1mA3hUrnblRq41TkolVl

[Edit script properties](#)

API: Generating Content

```
function generateContent() {
  const modelName = 'gemini-1.5-flash';
  const requestBody = {
    "contents": [
      {
        "parts": [
          {
            "text": promptGenerations('col')
          }
        ],
        "role": "user"
      }
    ]
  };

  const url = 'https://generativelanguage.googleapis.com/';
  const options = {
    "method": "POST",
    "contentType": "application/json",
    "payload": JSON.stringify(requestBody)
  };
  const response = UrIFetchApp.fetch(url, options);
  if (response.getResponseCode() === 200) {
    const responseData = JSON.parse(response.getContentText());
    const generatedText = responseData.candidates[0].content;
    console.log(generatedText);
  } else {
    console.error('Error calling Gemini API: ${response.status}');
  }
}
```

API Document

<https://ai.google.dev/api/generate-content>

The screenshot shows the Google AI for Developers API Reference page for the `models.generateContent` method. The page is titled "Generating content" and includes a navigation menu with options like "Overview", "API versions", "Capabilities", "Models", "Tokens", "Files", "Caching", "Embeddings", "Tuning", "Semantic retrieval", "All methods", and "Deprecated". The main content area is titled "Method: models.generateContent" and describes the method's purpose: "Generates a response from the model given an input GenerateContentRequest." It also provides the endpoint URL: `https://generativelanguage.googleapis.com/v1beta/(model=models/*):generateContent` and the path parameter `model` of type `string`. An example request is shown in a code block, demonstrating how to use the `generateContent` method in Python.

```
model = genai.GenerativeModel("gemini-1.5-flash")
response = model.generate_content("Write a story about a magic backpack.")
print(response.text)
```

Task 1: Prompt Generation – code:3.1

學習重點

- 各式 Prompt 技巧與範例
- 設計一個 Function 可以切換 Prompt

3.1 Prompt Generation

```
function promptGeneration(nums) {  
  //Explicit Instructions  
  //Role Prompting: You are a PhD in mathematics! ...  
  //In-Context Learning: Zero shot, One shot, Few shot  
  //Chain of Thought (CoT): Let's think step by step?  
  //Reduce Hallucination: Retrieval Augmented Generation (RAG)  
  
  const explicitInstructionsPrompt = "台灣有多少城市，首都在哪？氣候如何？"  
  const rolePrompting = "你是一位台灣資深地理學家，請問台灣有多少城市，首都在哪？氣候如何？"  
  const zeroShot = "該國所在的首都名稱是？該城市是不是該國最大城市？詢問內容：台灣的首都在哪？"  
  const oneShot = "南韓的首都是首爾，但他並不是最大的城市，詢問內容：台灣的首都？"  
  const chainOfThought = "台灣的首都是哪裡？請詳細解釋首都的特徵和其重要性。核心問題、基本知識、首都的特徵、首都的重要性、請幫我總結回答"  
  
  switch (nums) {  
    case 'explicitInstructionsPrompt':  
      return explicitInstructionsPrompt;  
    case 'rolePrompting':  
      return rolePrompting  
    case 'zeroShot':  
      return zeroShot;  
    case 'oneShot':  
      return oneShot;  
    case 'cot':  
      return chainOfThought;  
    default:  
      return explicitInstructionsPrompt  
  }  
}
```

Task 2: Token Count – code:3.2

學習重點

- 計算每一次呼叫所傳入的 Input or Output 共計多少 Tokens。
- 此 Tokens or Characters 會是 LLM 的計價基礎。

3.2 Token Counts

```
function countTokens() {
  const modelName = 'gemini-1.5-flash';
  const requestBody = {
    "contents": [
      {
        "parts": [
          {
            "text": promptGeneration('rolePrompting')
          }
        ],
        "role": "user"
      }
    ]
  };
  const url =
`https://generativelanguage.googleapis.com/v1/models/${modelName}:countTokens?key=${getGeminiApiKey()}`;
  const options = {
    "method": "POST",
    "contentType": "application/json",
    "payload": JSON.stringify(requestBody)
  };
  const response = UrlFetchApp.fetch(url, options);
  if (response.getResponseCode() === 200) {
    const responseData = JSON.parse(response.getContentText());
    console.log(responseData.totalTokens)
  } else {
    console.error(`Error calling Gemini API:
${response.getResponseCode()}`);
  }
}
```

Task 2: Generating Content – code:3.3

學習重點

- 主要執行內容生成的 API
- 認識參數以及角色
- 取得 Gemini Model 結果

3.3 Generate Content

```
function generateContent() {
  const modelName = 'gemini-1.5-flash';
  const requestBody = {
    "contents": [
      {
        "parts": [
          {
            "text": promptGeneration('col')
          }
        ],
        "role": "user"
      }
    ]
  };

  const url =
'https://generativelanguage.googleapis.com/v1/models/${modelName}:generateContent?key=${getGeminiApiKey()}';
  const options = {
    "method": "POST",
    "contentType": "application/json",
    "payload": JSON.stringify(requestBody)
  };

  const response = UrlFetchApp.fetch(url, options);
  if (response.getResponseCode() === 200) {
    const responseData = JSON.parse(response.getContentText());
    const generatedText =
responseData.candidates[0].content.parts[0].text
    console.log(generatedText)
  } else {
    console.error("Error calling Gemini API:
${response.getResponseCode()}");
  }
}
```

Task 4: Embedding – code:3.4

學習重點

- 學習什麼是 Embedding 以及用途為何?

3.4 Embedding

```
function embeddingContent() {
  const modelName = 'embedding-001';
  const requestBody = {
    "model": "models/embedding-001",
    "content": {
      "parts": [
        {
          "text": promptGeneration('explicitInstructionsPrompt')
        }
      ]
    }
  };
  const url =
`https://generativelanguage.googleapis.com/v1/models/${modelName}:emb
edContent?key=${getGeminiApiKey()}`;
  const options = {
    "method": "POST",
    "contentType": "application/json",
    "payload": JSON.stringify(requestBody)
  };
  const response = UrlFetchApp.fetch(url, options);
  if (response.getResponseCode() === 200) {
    const responseData = JSON.parse(response.getContentText());
    console.log(responseData)
  } else {
    console.error(`Error calling Gemini API:
${response.getResponseCode()}`);
  }
}
```

Task 5: List/Get Gen AI Model – code:3.5

學習重點

- 查閱現行 GCP 提供什麼 LLM Model
- 了解 Model 參數

3.5 List/Get Gen AI Model

```
function listModels() {
  const url =
`https://generativelanguage.googleapis.com/v1/models?key=${getGeminiA
piKey()}`;
  const options = {
    "method": "GET",
    "contentType": "application/json"
  };
  const response = UrlFetchApp.fetch(url, options);
  if (response.getResponseCode() === 200) {
    const responseData = JSON.parse(response.getContentText());
    console.log(responseData)
  } else {
    console.error(`Error calling Gemini API:
${response.getResponseCode()}`);
  }
}

function getSpecificModel() {
  const modelName = 'gemini-1.5-flash'
  const url =
`https://generativelanguage.googleapis.com/v1/models/${modelName}?key
=${getGeminiApiKey()}`;
  const options = {
    "method": "GET",
    "contentType": "application/json"
  };
  const response = UrlFetchApp.fetch(url, options);
  if (response.getResponseCode() === 200) {
    const responseData = JSON.parse(response.getContentText());
    console.log(responseData)
  } else {
```

04

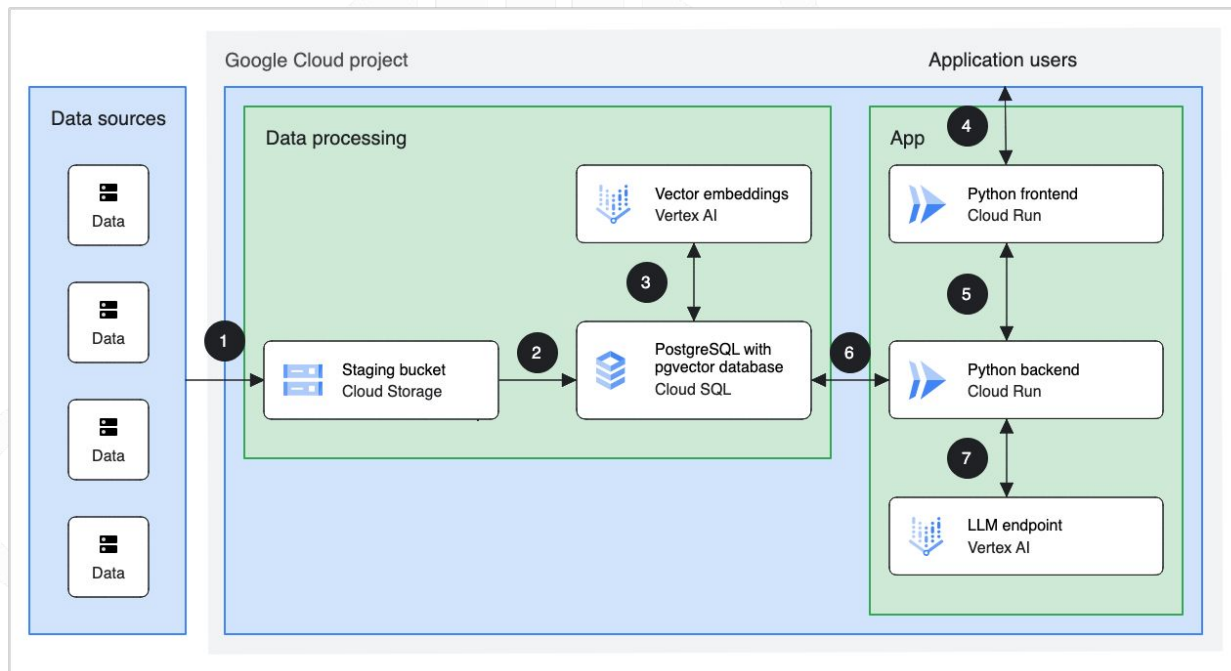
利用 RAG 技術，將 Gmail 內容提
供給 Gemini 做信件整理



CloudMile

Retrieval-Augmented Generation – RAG

RAG 是一種結合檢索和生成的架構，
可以從多個資料來源檢索內容並生成自然語言回答。



source:

<https://cloud.google.com/architecture/ai-ml/generative-ai-rag>

Gmail API – Code 4.1

學習重點

- 串接 Gmail API
- 取得 Subject 以及 Mail Body

4.1 Get Gmail Data

```
function queryGmail() {
  const threads = GmailApp.search('from:"email@gmail.com"', 0, 5);
  const datas = []
  console.log(threads.length)
  for(var i =0; i< threads.length; i ++){
    const id = threads[i].getId()
    const messages = threads[i].getMessages();
    const name = threads[i].getFirstMessageSubject()

    for (var j = 0; j < messages.length; j++) {
      Logger.log(messages[j].getPlainBody());
      datas.push(messages[j].getPlainBody());
    }
    Logger.log(id);
    Logger.log(name);
  }
  return datas;
}
```

RAG with Gemini API – Code 4.2

學習重點

- 理解 RAG 的效果並實作 RAG

4.2 RAG

```
function geminiRAG() {
  const allData = queryGmail();
  const mailContent = allData.join("\n\n");
  const summaryPrompt = "你是一位整理文件的專家，以下是五封郵件的內容，請提供這些郵件的整體摘要並條列式提醒我該注意的事項: " + "\n\n" + mailContent;
  const modelName = 'gemini-1.5-flash';
  const requestBody = {
    "contents": [
      {
        "parts": [
          {
            "text": summaryPrompt
          }
        ],
        "role": "user"
      }
    ]
  };

  const url =
'https://generativelanguage.googleapis.com/v1/models/${modelName}:generateContent?key=${getGeminiApiKey()}';
  const options = {
    "method": "POST",
    "contentType": "application/json",
    "payload": JSON.stringify(requestBody)
  };

  const response = UrlFetchApp.fetch(url, options);
  if (response.getResponseCode() === 200) {
    const responseData = JSON.parse(response.getContentText());
    const generatedText =
responseData.candidates[0].content.parts[0].text
    console.log(generatedText)
  } else {
```

05

總結：Integrate with AppScript
and Gemini API



CloudMile

Integrate with AppScript and Gemini API

- **了解大型語言模型（以 Gemini 為例）**
 - 了解各個大型語言模型的參數及功能
- **學習呼叫 Gemini API**
 - 透過 AppScript 串接 Gemini API
 - 學習各式 Gemini API 的使用情境
- **熟悉 Prompt Engineering 以及 RAG**
 - 熟悉生成式 AI 的應用技巧

課後問卷連結



**Thank
You!**