# **GWS Apps Script Gemini Sample Code**

## Appsscript.json

{

"timeZone": "Asia/Taipei",

"exceptionLogging": "STACKDRIVER",

"runtimeVersion": "V8",

"dependencies": {

},

"oauthScopes": [

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

"https://www.googleapis.com/auth/userinfo.email",

"https://www.googleapis.com/auth/script.external\_request"

],

"webapp": {

"executeAs": "USER\_DEPLOYING",

"access": "ANYONE\_ANONYMOUS"

}

}

## Properties

### 2.1 getGeminiApiKey

function getGeminiApiKey() {

const scriptProperties = PropertiesService.getScriptProperties();

const geminiApiKey = scriptProperties.getProperty('GEMINI\_API\_KEY');

console.log(geminiApiKey);

return geminiApiKey;

}

## Gemini API

### 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

}

}

### 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()}`);

}

}

### 3.3 Generate Content

function generateContent() {

const modelName = 'gemini-1.5-flash';

const requestBody = {

"contents": [

{

"parts": [

{

"text": promptGeneration('explicitInstructionsPrompt')

}

],

"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()}`);

}

}

### 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/${modelName}:embedContent?key=${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()}`);

}

}

### 3.5 List/Get Gen AI Model

function listModels() {

const url = `https://generativelanguage.googleapis.com/v1/models?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 {

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 {

console.error(`Error calling Gemini API: ${response.getResponseCode()}`);

}

}

## RAG

### 4.1 Get Gmail Data

function queryGmail() {

const threads = GmailApp.search('from:"{YOUR\_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;

}

### 

### 

### 

### 

### 

### 4.2 RAG

function geminiRAG() {

const allData = queryGmail();

const mailCotent = 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" + mailCotent;

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 {

console.error(`Error calling Gemini API: ${response.getResponseCode()}`);

}

}