How to Built AI Tools with Google Sheets by Fabian Markl

Links

- MailMimic AI
- Set up a Google Account
- Set up ChatGPT Account
- Set up OpenAI account (to get API Key)
- Setup Claude.ai account
- Setup Anthropic account (to get API Key)
- OpenAI API / GPT-4o Documentation
- Anthropic API / Claude Documentation

Google App Scripts

- Send Slides as PDF via Email - p1
- Send Slides as PDF via Email - p2

Prompts

- Ask if Problem Can Be Solved With a Google Sheet
- Generate App Script for Certain Functionality
- Fixing Error in Function from Logging
- Let it Explain each part of function do you
- API Connection GPT-4o
- API Connection Claude Sonnet 3.5
- Add “Install” to Custom Menu
- Make AI tool More Valuable

Links

MailMimic AI

mailmimic.com

Fabian’s YouTube Channel

youtube.com/@fabimarklAI

Set up a Google Account:

https://accounts.google.com/

Set up ChatGPT Account:

https://chatgpt.com/auth/login

Set up OpenAI account (to get API Key):

https://platform.openai.com/signup

Setup Claude.ai account:

https://claude.ai/

Setup Anthropic account (to get API Key):

https://console.anthropic.com/

OpenAI API / GPT-4o Documentation

https://platform.openai.com/docs/models/gpt-4o
https://platform.openai.com/api-keys

Anthropic API / Claude Documentation

https://docs.anthropic.com/en/docs/about-claude/models https://console.anthropic.com/settings/keys

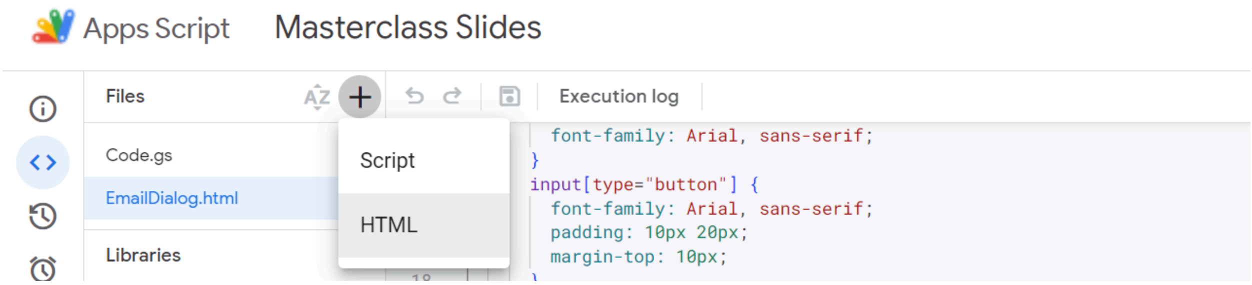
Google App Scripts

Send Slides as PDF via Email - p1

```
function onOpen() {
  var ui = SlidesApp.getUi();
  ui.createMenu('Custom Actions')
    .addItem('Send Slides', 'showEmailDialog')
    .addToUi();
}

function showEmailDialog() {
  var htmlOutput = HtmlService
      .createHtmlOutputFromFile('EmailDialog')
      .setWidth(400)
      .setHeight(250);
  SlidesApp.getUi().showModalDialog(htmlOutput, 'Who wants the slides as
PDF?');
}

function emailPresentationAsPDF(emails) {
  try {
    var presentation = SlidesApp.getActivePresentation();
    var presentationId = presentation.getId();
    var presentationName = presentation.getName();
    var file = DriveApp.getFileById(presentationId);
    var pdfBlob = file.getAs('application/pdf');

    var subject = 'Slides: ' + presentationName;
    var body = 'Hey,\n\nI\'ve attached the slides as PDF of your presentation
' + presentationName + '.\n\n- YOUR NAME';

    var emailList = emails.split(/[,;\n]/).map(function(email) {
      return email.trim();
    }).filter(function(email) {
      return email.length > 0;
    });

    emailList.forEach(function(emailAddress) {
      MailApp.sendEmail({
        to: emailAddress,
        subject: subject,
        body: body,
        attachments: [pdfBlob],
        name: 'EMAIL-SENDER-NAME'
      });
    });

    Logger.log('Email sent to: ' + emailList.join(', '));
  } catch (error) {
    Logger.log('Error sending email: ' + error.message);
  }
}
```

Send Slides as PDF via Email - p2

Click on “+” to add a file and select HTML -> Name file “EmailDialog”



```
  font-family: Arial, sans-serif;
}
input[type="button"] {
  font-family: Arial, sans-serif;
  padding: 10px 20px;
  margin-top: 10px;
```

```
<!DOCTYPE html>
<html>
  <head>
    <base target="_top">
    <style>
      body {
        font-family: Arial, sans-serif;
      }
      textarea {
        width: 100%;
        height: 80px; /* Adjusted height to fit the dialog */
        font-family: Arial, sans-serif;
      }
      input[type="button"] {
        font-family: Arial, sans-serif;
        padding: 10px 20px;
        margin-top: 10px;
      }
    </style>
  </head>
  <body>
    <form>
      <label for="emails">Enter email addresses (separated by new
lines):</label><br>
      <textarea id="emails" name="emails" rows="4" cols="50"></textarea><br>
      <input type="button" value="Send Slides" onclick="sendEmails()">
    </form>
    <script>
      function sendEmails() {
        var emails = document.getElementById('emails').value;

google.script.run.withSuccessHandler(onSuccess).withFailureHandler(onFailure)
.emailPresentationAsPDF(emails);
        google.script.host.close();
      }

      function onSuccess() {
        alert('Emails sent successfully!');
      }

      function onFailure(error) {
        alert('Failed to send emails: ' + error.message);
      }
    </script>
  </body>
</html>
```

Prompts

Ask if Problem Can Be Solved With a Google Sheet

Is it possible to create Google Apps Script for a Google Sheet that can do the following:

[Insert what you would like the tool to do]

Generate App Script for Certain Functionality

I need a google apps script that does this:

[List of functions/tasks to be performed]

Make sure to include detailed comments and logging in the code so I understand how it works.

Fixing Error in Function from Logging

I ran the function and in the log I got this error, please fix it:
[Insert Error Log]

Only give me the code snippets I really need to remove, update or replace and tell me exactly where to do that. Do not give me the full code.

Let it Explain each part of function do you

Please explain to me a non-techy person what each part of the following function does:

[Insert Code of Function]

API Connection GPT-4o

I'm going to teach you how to make the correct API Call for the specific AI model I want to use in future functions. Just understand it and then maintain its exact structure and make the adjustments I will tell you:

```
function analyzeAllExistingEntries() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var lastRow = sheet.getLastRow();

  // Setup for OpenAI API with the chat model
  var apiURL = 'https://api.openai.com/v1/chat/completions';  // Correct
endpoint for chat models
  var apiKey = 'YOUR-API-KEY';  // Your secured API key
  var headers = {
    "Content-Type": "application/json",
    "Authorization": "Bearer " + apiKey
  };

  // Loop through all rows that have data in column E and column D
  for (var row = 2; row <= lastRow; row++) {
    var textE = sheet.getRange(row, 5).getValue();  // Text from Column E
    var textD = sheet.getRange(row, 4).getValue();  // Text from Column D
    var combinedText = textE + " " + textD;  // Combine text from Column E
and D

    if (combinedText.trim() !== '') {  // Ensure the combined text is not
empty
      // Enhanced prompt for a detailed analysis of enjoyment, interest, and
emotions
      var prompt = `As an expert in analyzing descriptions of activities,
provide a detailed analysis in one or two short but very specific sentences.
Focus on identifying what specific activities, tasks and topics the person
enjoys or finds interesting. And also specific tools he likes to use. #Rules:
You must include specific details. Example: Don't say "Social Media" when the
person mentions Instagram, Twitter or LinkedIn. You MUST the mention the
specific platform or tool. Again, be as specific as possible with all topics,
tasks and tools the person described. Plus describe the key emotions
expressed. The following is the experience the person has described that you
need to perform this analysis with: "${combinedText}"`;

      var payload = JSON.stringify({
        "model": "gpt-4o",  // Specific model you mentioned
        "messages": [
          {"role": "system", "content": prompt},
          {"role": "user", "content": combinedText}
        ],
        "max_tokens": 150
      });
      var options = {
        "method": "post",
        "headers": headers,
        "payload": payload
      };

      // Fetch the response from OpenAI
      var response = UrlFetchApp.fetch(apiURL, options);
      var responseJson = JSON.parse(response.getContentText());
      var analysis = responseJson.choices[0].message.content.trim();

      // Write the results back to the Google Sheet in column I
      sheet.getRange(row, 9).setValue(analysis);  // Column I for results
    }
  }
}
```

API Connection Claude Sonnet 3.5

I’m going to teach you how to make the correct API Call for the specific AI model I want to use in future functions. Just understand it and then maintain it’s exact structure and make the adjustments I will tell you:

```
function analyzeEntriesWithClaude() {
  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();
  var lastRow = sheet.getLastRow();

  var apiURL = 'https://api.anthropic.com/v1/messages';
  var apiKey = 'YOUR-API-KEY'; // Replace with your actual Anthropic API key
  var headers = {
    "Content-Type": "application/json",
    "x-api-key": apiKey,
    "anthropic-version": "2023-06-01"
  };

  Logger.log('Starting analysis with Claude 3.5 Sonnet');

  for (var row = 2; row <= lastRow; row++) {
    var textE = sheet.getRange(row, 5).getValue();
    var textD = sheet.getRange(row, 4).getValue();
    var combinedText = textE + " " + textD;
    if (combinedText.trim() !== '') {
      Logger.log('Analyzing row ' + row);

      var prompt = `As an expert in analyzing descriptions of activities,
provide a detailed analysis in two brief but very specific sentences. Focus
on identifying what specific activities, tasks and topics the person enjoys
or finds interesting. And also specific tools he likes to use. #Rules: You
must include specific details. Example: Don't say "Social Media" when the
person mentions Instagram, Twitter or LinkedIn. You MUST the mention the
specific platform or tool. Again, be as specific as possible with all topics,
tasks and tools the person described. Plus describe the key emotions
expressed. The following is the experience the person has described that you
need to perform this analysis with: "${combinedText}"`;

      var payload = JSON.stringify({
        "model": "claude-3-5-sonnet-20240620",
        "messages": [
          {"role": "user", "content": prompt}
        ],
        "max_tokens": 150
      });

      var options = {
        "method": "post",
        "headers": headers,
        "payload": payload,
        "muteHttpExceptions": true
      };

      try {
        var response = UrlFetchApp.fetch(apiURL, options);
        var responseJson = JSON.parse(response.getContentText());

        // Log the full response for debugging
        Logger.log('API Response: ' + JSON.stringify(responseJson));

        // Extract the analysis from the response
        var analysis = responseJson.content[0].text.trim();

        sheet.getRange(row, 10).setValue(analysis);
        Logger.log('Analysis for row ' + row + ' completed');
      } catch (error) {
        Logger.log('Error in row ' + row + ': ' + error);
        Logger.log('Full error object: ' + JSON.stringify(error));
        sheet.getRange(row, 10).setValue('Error: ' + error);
      }
```

Add “Install” to Custom Menu

I need a Google App Script that does this:

- has a function that sets up all my manual triggers that I will tell you.

- It creates a custom menu called “Setup” with option called “Install” which then runs the function you created to set up all my manual triggers:

Here are my manual triggers:

[Copy and paste list of triggers]

I already have this Custom Menu

[Insert Custom Menu Code “OnOpen”]

Make AI tool More Valuable

My AI tool that I built using [Insert Tools you utilize, like GSheets, APIs etc.] which does this:

[List what your tool does]

Give me 3 ideas for each of the following that I can implement in my AI tool to:

A) Get user better results

B) Remove steps for user

C) Save user more time

Think outside the box, what else a user of the core function of my tool would also benefit from. You can use any of the tools that I already use in my Tool as well as other Google Workspace apps, or Software that have API access.