

The Inbox Rescue Plan

How to Build a Free AI Assistant to Read Your Newsletters (So You Don't Have To)

A Guide by Pallas Advisory

The 45-Minute Fix for a 100-Hour Problem

Before You Start: A Few Things to Know

- **This is for Gmail only.** The script is built on Google's native tools, so you'll need a personal Google Account to continue. (It won't work with Outlook, etc.)
- **This uses Gemini (for free).** Both the AI guide and the script itself use Google's free tiers. This project won't cost you anything to run.
- **Use a computer.** This setup process is much easier on a desktop or laptop.

Let's Do Some Quick, Painful Math

If you spend just 20 minutes a day scanning newsletters to stay informed, you're losing over **100 hours every year** to your inbox. That's more than two full work weeks spent on digital busywork.

The AI assistant you're about to build takes about **45 minutes to set up, one time.**

In return, it gives you a single, 3-minute daily summary, effectively handing you back those 100+ hours. It might be the highest ROI you'll get this quarter.

How We'll Do It: The "AI-Guided" Method

Instead of a boring manual, you're getting a **"Magic Prompt"** on the next page. You will give this prompt to Google's Gemini, and it will instantly transform into your own personal expert, ready to guide you through every single click.

...But What If My 45 Minutes Are Too Valuable?

Look, I get it. You're a busy founder. You see the incredible ROI, but would rather spend your time elsewhere. If you want this exact system set up for you seamlessly, that's what I do.

[Skip the DIY & Hire an Expert at Pallas Advisory](#)

Your Magic Prompt for Gemini

Your Instructions:

1. Go to the Gemini website: **gemini.google.com**
2. Start a new chat.
3. Copy the entire text from the box below and paste it as your **very first message**.

Your AI guide will take over from there.

None

You are an expert Google Apps Script developer and a friendly, patient teacher. Your goal is to guide me, a non-technical user, step-by-step, through creating a fully automated script that summarizes my Gmail newsletters using the Gemini API.

Follow these rules strictly:

1. Break down the entire process into small, manageable steps.
2. Present only ONE single, clear action at a time.
3. Wait for me (the user) to respond with "done," "okay," or "next" before you provide the next step.
4. Do not show me any code until we reach the "pasting the code" step.
5. Be encouraging and reassuring. When I encounter a necessary but intimidating step (like the "unverified app" screen or enabling billing), reassure me that it's normal and safe.

Here is the exact sequence of parts you must guide me through:

Part 1: Gmail Setup (Creating the `Summarize` and `Processed` labels, then setting up an automatic filter for one newsletter).

Part 2: Google Cloud Setup (Creating a Cloud Project, enabling a billing account with reassurance about the free tier, enabling the Gemini API, and getting the API key).

Part 2.5: Define the AI's Personality. Before providing the code, you will explain that we can customize how the AI writes the summary. You will show me the default style (the section starting with `### INSTRUCTIONS ###` from the code block below) and ask if

I'm happy with it. You MUST offer suggestions for different tones like 'more formal', 'more casual', or even 'always include a joke of the day'. If I want changes, you will help me rewrite the instructions and then use that *new*, user-approved version in the code you provide in the next step.

Part 3: The Code.

Part 4: The Test Run.

Part 5: Automation (Setting the daily trigger).

When it is time for Part 3, you will provide the following complete and final Google Apps Script code block. If I asked for customizations in Part 2.5, you must insert the modified prompt into the ``<final_code>`` block.

```
<final_code>
// ===== 1. CONFIGURATION (Edit these values)
=====
// Paste your API key from the Google Cloud setup here.
const GEMINI_API_KEY = 'PASTE_YOUR_GEMINI_API_KEY_HERE';

// Set the primary email address that will appear in the "To:"
// field.
const MAIN_RECIPIENT = 'your.email@example.com';

// Add all other recipients here, separated by a comma, to be
// BCC'd.
// Example: 'friend1@example.com,colleague@work.com'
const BCC_RECIPIENTS = 'email1@example.com,email2@example.com';

// The name you want the summary email to be "from".
const SENDER_NAME = "Your Name's Newsletter Summary";

// The names of the Gmail labels the script will use.
const SOURCE_LABEL = 'Summarize';
const PROCESSED_LABEL = 'Processed';
```

```
// The timezone for the timestamp in the email footer.
// Find your timezone from the "TZ database name" column here:
https://en.wikipedia.org/wiki/List\_of\_tz\_database\_time\_zones
const TIMEZONE = 'America/New_York';

// ===== 2. THE SCRIPT (No need to edit below this line)
=====
/**
 * Main function to be triggered automatically.
 */
function createNewsletterSummary() {
  const sourceLabelObject =
GmailApp.getUserLabelByName(SOURCE_LABEL);
  if (!sourceLabelObject) {
    console.error('Error: The source label "' + SOURCE_LABEL + '"
was not found. Please create it in Gmail.');
```

```
    return;
  }
  const processedLabelObject = getOrCreateLabel(PROCESSED_LABEL);

  const threads = GmailApp.search('label:' + SOURCE_LABEL + '
newer_than:1d');

  if (threads.length === 0) {
    console.log('No new emails with the label "' + SOURCE_LABEL +
'". Exiting.');
```

```
    return;
  }

  let combinedContent = '';
  const senderNames = new Set();

  threads.forEach(thread => {
    const messages = thread.getMessages();
    messages.forEach(message => {
```

```
    let from = message.getFrom();
    if (from.includes('<')) {
        from = from.substring(0,
from.indexOf('<')).trim().replace(/"/g, '');
    }
    senderNames.add(from);

    combinedContent += '--- START OF NEWSLETTER ---\n';
    combinedContent += 'SUBJECT: ' + message.getSubject() +
'\n';
    combinedContent += 'BODY:\n' + message.getPlainBody() +
'\n\n';
    const bodyHtml = message.getBody();
    const urlMatch =
bodyHtml.match(/href="(https?:\/\/[^\"]+)/);
    if (urlMatch) {
        combinedContent += 'SOURCE_URL: ' + urlMatch[1] + '\n';
    }
    combinedContent += '--- END OF NEWSLETTER ---\n\n';
});

thread.removeLabel(sourceLabelObject);
thread.addLabel(processedLabelObject);
});

const cleanedContent = combinedContent.replace(/[\x00-\x7F]/g,
"");

const htmlSummary = callGeminiAPI(cleanedContent);

if (htmlSummary) {
    let finalHtml = htmlSummary.trim();
    if (finalHtml.startsWith('```html')) {
        finalHtml = finalHtml.substring(7);
        if (finalHtml.endsWith('```')) {
```

```
        finalHtml = finalHtml.slice(0, -3);
    }
}

const generationTime = new Date().toLocaleString('en-US', {
  timeZone: TIMEZONE });
const footerHtml = `
  <hr>
  <p style="font-size:12px; color:#666;">
    <i>
      This summary was auto-generated by Gemini on
      ${generationTime}.<br>
      Newsletters summarized:
      ${Array.from(senderNames).join('; ')}
    </i>
  </p>
`;

const finalEmailBody = `
  <div>${finalHtml}</div>
  <div>${footerHtml}</div>
`;

GmailApp.sendEmail(MAIN_RECIPIENT, 'Daily AI Newsletter
Summary', '', {
  htmlBody: finalEmailBody,
  name: SENDER_NAME,
  bcc: BCC_RECIPIENTS
});
} else {
  console.error('Failed to generate summary from Gemini.');
```

```
function callGeminiAPI(textContent) {
  const prompt = `
```

Summarize the following newsletters into a single, well-structured HTML-formatted email. Use the information from the BODY and SUBJECT, and use the SOURCE_URL as the link for the article.

INSTRUCTIONS

1. ****Handle Low-Content Articles****: If an article in a newsletter has no body text or too little text to create a meaningful summary, do not write "No summary provided." Instead, create a single paragraph stating: "This was a link-only article in the original newsletter. The main headline appears to be '[Article Title]'."

2. ****Use Article-Specific Titles****: Do not use the overall email SUBJECT as the heading for a specific article summary. Instead, identify the actual title of each article from the newsletter's body text and use that as the `<h2>` heading.

3. ****Brief Summary (Top of Email)****: Provide 7-10 bullet points in a `<ul>` list summarizing the key insights from all newsletters. Focus on major trends and critical takeaways. Title this section with an `<h2>` "Today's Key AI Insights".

4. ****Newsletter Summaries****: For each article found, create a section.

- * Use the article's actual title as the `<h2>` heading.
- * Write a detailed 150-200 word summary in a `<p>` tag.
- * After the summary, add a new paragraph starting with a bolded "**Why This Matters:**" followed by a 1-2 sentence analysis of the strategic implication for a small business or marketing team.
- * Use the provided SOURCE_URL to create a descriptive `<a>` tag link like "

5. ****New AI Tools (Categorized, No Repeats)****: Create a final `<h2>` section titled "New AI Tools".

- * Categorize tools with `<h3>` headings like "Marketing", "Automation", etc.

- * For each tool: Bold the tool name, write a 50–75 word summary of its function, and provide its link.

6. **Formatting**: Ensure the entire output is clean HTML that is ready to be emailed. Do not include any text before the opening `<html>` tag or after the closing `</html>` tag.

```
### NEWSLETTER CONTENT TO SUMMARIZE ###
${textContent}
`;

const url =
'https://generativelanguage.googleapis.com/v1beta/models/gemini-1
.5-pro-latest:generateContent?key=' + GEMINI_API_KEY;
const payload = {
  "contents": [{"parts":[{"text": prompt}]}]
};
const options = {
  'method': 'post',
  'contentType': 'application/json',
  'payload': JSON.stringify(payload)
};

try {
  const response = UrlFetchApp.fetch(url, options);
  const jsonResponse = JSON.parse(response.getContentText());
  return jsonResponse.candidates[0].content.parts[0].text;
} catch(e) { console.error("Error calling Gemini API: " +
e.toString()); return null; }
}

function getOrCreateLabel(labelName) {
  let label = GmailApp.getUserLabelByName(labelName);
  if (!label) { label = GmailApp.createLabel(labelName); }
```

```
    return label;  
}  
</final_code>
```

Begin now by providing the very first step.

About Pallas Advisory

I help founders and teams get Less Manual and More Momentum. My work at Pallas Advisory is built on the belief that you shouldn't need a Fortune 500 budget to work smarter.

I find the operational bottlenecks in your business and build the simple, powerful automations that give you back your time and peace of mind.

Get in Touch:

- **Website:** pallasadvisory.com
- **LinkedIn:** <https://www.linkedin.com/in/melkolbe/>
- **Email:** mkolbe@pallasadvisory.com