## Your Guide to Deploying Google Apps Script Automation Systems

Welcome! This document will walk you through the essential steps to get your new Google Apps Script automation system up and running. Follow these instructions carefully to ensure a smooth deployment.

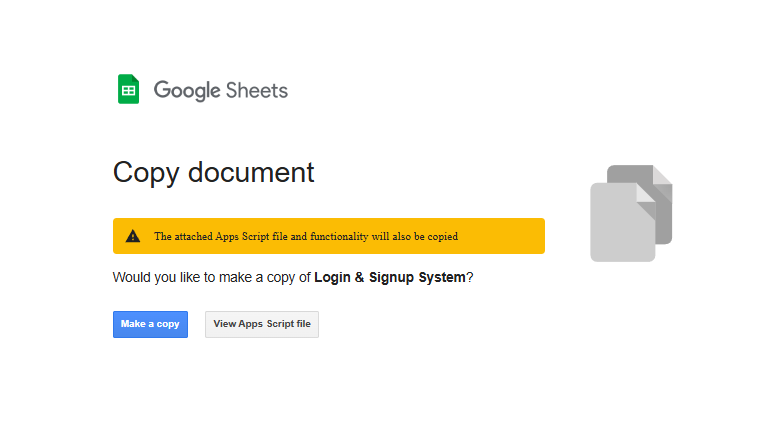
**Goal:** To help you copy, configure, and activate your automation system within your own Google Workspace environment.

### 1. Making a Copy of Your Automation System

Before you can use any automation, you need to create your own copies of the script and any associated files. This ensures the system runs independently within your Google account and doesn't affect the original template.

#### 1.1. Copying the Google Apps Script Project

This is the core of your automation.

1. **Open the Script Container Link:** Click on the "Script Container" URL provided in the email you received.
2. **Make a Copy:**
   * Once the file loads, select **Make a copy**.
   * 
   * A new sheet project will open in a new tab. This is your personal copy.
   * **Rename Your Copy:** It's highly recommended to rename your copied sheet immediately. Give it a meaningful name (e.g., "My Gmail Autoresponder" or "Budget Tracker Automation"). This makes it easier to manage later.

#### 1.2. Copying Associated Google Sheets/Docs/Folders (if applicable)

Many automation systems work with Google Sheets, Docs, or other files. You'll need your own copies of these too.

1. **Open the Folder Link:** If your email included a "Folder" URL, click it to open the Google Drive folder containing the associated files.
2. **Select All Files:** Press Ctrl + A (Windows) or Cmd + A (Mac) to select all items in the folder.
3. **Make a Copy:** Right-click on the selected files and choose **"Make a copy"**.
4. **Organize Your Copies:** Google Drive will create "Copy of..." versions in the same folder. It's best to:
   * **Move them:** Select all the "Copy of..." files, right-click, and choose **"Move to"**. Create a new, dedicated folder in your Google Drive (e.g., "My Automation Projects") and move them there.
   * **Rename them:** Rename each "Copy of..." file to something more descriptive (e.g., "My Budget Sheet", "My Template Doc").

### 2. Initial Setup and Prerequisites

Once you have your copies, you might need to connect them or enable specific services.

#### 2.1. Linking Your Script to Your Copied Files

* **Check the Script:** Open your copied Google Apps Script project. Look for comments or variables at the top of the Code.gs file (or other .gs files) that mention SPREADSHEET\_ID, DOCUMENT\_ID, FOLDER\_ID, or similar.
* **Update IDs:** Replace the placeholder IDs/URLs with the actual IDs of *your copied* Google Sheets, Docs, or Folders.
  + **How to get an ID:** The ID is a long string of characters found in the URL of your Google Sheet/Doc/Folder. For example, in https://docs.google.com/spreadsheets/d/YOUR\_SPREADSHEET\_ID/edit, YOUR\_SPREADSHEET\_ID is what you need.

#### 2.2. Enabling Google Services/APIs (if required)

Some scripts require access to advanced Google services (like Gmail API, Calendar API, Drive API).

1. **In the Apps Script Editor:** On the left sidebar, click **"Services"** (the + icon).
2. **Find and Add:** Browse the list of services. If the script's instructions or comments mention a specific API (e.g., "Please enable the Drive API"), find it in the list, click **"Add"**, and then **"Add"** again in the pop-up.
3. **Authorization:** The first time your script runs and tries to use an enabled service, Google will ask you to authorize it. Follow the on-screen prompts.

### 3. Deploying the Script as a Web App (if applicable)

If your automation system includes a user interface (like a form or a dashboard) accessible via a web link, you need to deploy it as a Web App.

1. **In the Apps Script Editor:** Go to **Deploy > New deployment**.
2. **Select Deployment Type:** Click on **"Select type"** and choose **"Web app"**.
3. **Configuration:**
   * **Execute as:** Set this to **"Me"** (your email address). This ensures the script runs with your permissions to access your sheets, send emails, etc.
   * **Who has access:** Set this to **"Anyone"**. This makes your web app publicly accessible via its URL. (If your script is for internal use only, you might choose "Anyone with Google account" or specific users).
   * **Description:** (Optional but Recommended) Add a short description for your deployment version.
4. **Deploy:** Click **"Deploy"**.
5. **Authorization:** The first time you deploy, you'll be prompted to review permissions. Click **"Review permissions"**, select your Google account, and click **"Allow"** to grant the necessary access.
6. **Get Web App URL:** After successful deployment, a dialog will appear with your **Web app URL**. Copy this URL. This is the link you'll use to access your automation system's web interface.
7. **Update Deployment for Changes:** Every time you make changes to your Code.gs (or index.html, script.js), you must create a **new deployment version** for the changes to take effect. Go to **Deploy > Manage deployments**, click the pencil icon next to your deployment, then **"Deploy"** again.

### 4. Setting Up Triggers (if applicable)

Some automation systems run automatically based on events (like a form submission, a spreadsheet edit) or at scheduled times. These are set up using Apps Script Triggers.

1. **In the Apps Script Editor:** On the left sidebar, click **"Triggers"** (the clock icon).
2. **Add Trigger:** Click the **"Add Trigger"** button in the bottom right corner.
3. **Configure Your Trigger:**
   * **Choose function to run:** Select the function name that needs to be triggered (e.g., onFormSubmit, processEmails, runDailyReport). The script's documentation or comments will indicate which function needs a trigger.
   * **Choose deployment to run:** Select your deployed Web App project (usually the only option here if you've deployed it).
   * **Select event source:**
     + **From spreadsheet:** For onEdit, onChange, onFormSubmit (if linked to a Google Form, though in your web app, this is handled by google.script.run).
     + **Time-driven:** For scripts that need to run daily, hourly, etc.
     + **From form:** If you used a Google Form directly.
     + **From calendar:** If the script interacts with Google Calendar events.
   * **Select event type:** (e.g., "On open", "On edit", "On form submit", "Day timer", "Hour timer").
   * **Failure notification settings:** Set how often you want to be notified of trigger failures (e.g., "Notify me immediately" or "Notify me daily").
4. **Save:** Click **"Save"**.
5. **Authorization (if new trigger):** If this is the first time setting up a trigger or if you add a new type of trigger, Google may ask for authorization again.

### 5. Troubleshooting Tips

If your automation system isn't working as expected, don't worry! Here are some common issues and how to resolve them:

1. **Permissions/Authorization Errors:**
   * **Issue:** "Authorization required" or script fails with permission errors.
   * **Solution:** Ensure you've granted all necessary permissions when prompted. Sometimes, running a function directly from the Apps Script editor (e.g., by selecting it in the dropdown and clicking "Run" (play icon)) can re-trigger the authorization prompt.
2. **Web App Not Working/Changes Not Appearing:**
   * **Issue:** Your web app doesn't load, or recent code changes aren't reflected.
   * **Solution:**
     + **New Deployment Version:** Remember, for code changes to appear in a deployed web app, you *must* create a **new deployment version** (Deploy > Manage deployments > Edit icon > New version > Deploy).
     + **Browser Cache:** Sometimes, your browser caches the old version. Try clearing your browser's cache or opening the web app in an Incognito/Private window.
3. **Script Not Running Automatically (Triggers):**
   * **Issue:** Your script doesn't run at the scheduled time or on a specific event.
   * **Solution:**
     + **Check Triggers:** Go to **Triggers** (clock icon) in the Apps Script editor. Verify that your trigger is correctly configured, pointing to the right function and deployment, and that its event source and type are correct.
     + **Check Execution Logs:** Go to **Executions** (list icon) in the Apps Script editor. This log shows all attempts to run your script (manual or by trigger) and any errors. This is your most valuable debugging tool!
4. **"Sheet/Doc/Folder not found" or "ID not found" Errors:**
   * **Issue:** The script can't find the Google Sheet, Document, or Folder it's supposed to interact with.
   * **Solution:** Double-check that the Spreadsheet ID, Document ID, or Folder ID you entered in your script's code is correct and belongs to *your copied* files. Ensure the sharing permissions of those files are set correctly for the script to access them.
5. **API Not Enabled Errors:**
   * **Issue:** "Service invoked with illegal or invalid parameters. (line X, file Y)" or similar cryptic errors, often related to services like "GmailApp", "DriveApp".
   * **Solution:** Go to **Services** (+ icon) in the Apps Script editor and ensure all necessary Google APIs mentioned in the script's instructions are enabled.
6. **Script Exceeding Execution Time/Memory Limits:**
   * **Issue:** For very large datasets or complex operations, scripts might hit Google's execution limits.
   * **Solution:** This is less common for standard automations. If it happens, you might need to optimize the script or break down the task into smaller, manageable chunks that run periodically.

#### How to Debug: Using Execution Logs

The **Executions** log in the Apps Script editor is your best friend for debugging.

* It lists every time your script has attempted to run.
* It tells you if the execution was successful, failed, or timed out.
* If it failed, you can click on the row to see the exact error message and the line number where the error occurred. This is crucial for identifying problems.

We hope this guide helps you successfully deploy your new automation system! If you encounter persistent issues after following these steps and checking the execution logs, please refer to the contact options provided on our portal for further assistance.