Google-Sheets-to-Gmail-Automation-Project

Project Overview

This automation workflow is designed to manage and distribute daily interview questions, likely for a study group, coding bootcamp, or personal learning initiative. It automates the process of:

1.Searching for pending questions in a Google Sheet.

2.Updating the status of processed questions.

3.Sending an email with the question and answer to specified recipients.

4.Scheduling the process to run daily at 6:00 AM.

5.The workflow uses Make.com (formerly Integromat), a no-code/low-code automation platform, to integrate Google Sheets and Gmail.

This eliminates manual work, ensuring questions are delivered consistently without human intervention.

Features

The project includes the following key features:

1.Automated Row Filtering: The workflow scans a Google Sheet named "JavaQuest" for rows where the Status column (column C) is marked as "Pending". It retrieves data from relevant columns (e.g., Question in column A, Answer in column B, and potentially other columns up to CZ).

2.Status Update: Once a question is processed, the workflow updates the Status column to "Sent", preventing the same question from being sent again.

3.Email Notification: The workflow sends an HTML-formatted email containing the question and answer to specified recipients. The email has a fixed subject: "Javaquest Daily Interview Question". The email is sent using a Gmail account.

4.Scheduled Execution: The automation runs daily at 6:00 AM, ensuring recipients receive a new question each morning.

Workflow Details

Here’s a step-by-step breakdown of how the automation works:

1.Search Rows Purpose: Identify the next question to send. Tool: Google Sheets module in Make.com. Details: Connects to a Google Sheet named "JavaQuest" (specifically, Sheet1). Uses a filter to find rows where column C (Status) is "Pending". Retrieves data from columns A (Question), B (Answer), and potentially other columns up to CZ (depending on the configuration). The connection is established using a Google account (devisrinivaskandikattu03@gmail.com). Why this matters: This step ensures only unprocessed questions are selected, maintaining a clean workflow.

2.Update a Row Purpose: Mark the processed question as completed. Tool: Google Sheets module in Make.com. Details: Updates the Status column (column C) to "Sent" for the row identified in the search step. Uses the row number from the previous step to ensure the correct row is updated. Uses the same Google Sheet connection as the search step. Why this matters: Updating the status prevents duplicate emails and keeps the sheet organized.

3.Send an Email Purpose: Deliver the question and answer to recipients. Tool: Gmail module in Make.com. Details: Sends an email to two recipients: devisrinivaskandikattu03@gmail.com Subject line: "Javaquest Daily Interview Question". Body: An HTML-formatted email that displays the question (from column A) and answer (from column B) in a clean, readable format. Uses a restricted Google connection linked to 21kb1a0566@nbkrist.org. Why this matters: The formatted email ensures recipients receive the content in an engaging, professional way.

4.Scheduling Purpose: Automate the process to run daily. Details: Configured in Make.com to execute the entire workflow at 6:00 AM every day. Ensures consistency without manual intervention. Why this matters: Scheduling eliminates the need for daily manual triggers, making the process fully autonomous.

Setup Instructions To replicate or deploy this automation, follow these steps:

Prerequisites

Google Account: You need a Google account with access to Google Sheets and Gmail. Ensure you have permissions to create and edit Google Sheets and send emails via Gmail. Make.com Account: Sign up for a Make.com account (or use another automation platform like Zapier, though the configuration file is specific to Make.com). A free tier may suffice for small workflows, but check Make.com's pricing for higher usage needs.

Google API Credentials: Set up Google API credentials for Google Sheets API and Gmail API via the Google Cloud Console. This involves creating a project, enabling the APIs, and generating OAuth credentials for Make.com to authenticate your Google account. Configuration

Create the Google Sheet: Create a Google Sheet named "JavaQuest". Structure the sheet with at least three columns: Column A: Question (e.g., "What is polymorphism in Java?") Column B: Answer (e.g., "Polymorphism allows objects to be treated as instances of their parent class…") Column C: Status (e.g., "Pending" or "Sent") Note the spreadsheet ID from the URL (e.g.,

https://docs.google.com/spreadsheets/d/1OsXTACsAcf0T6prPSg4DKoJOOw_JzEMd-olLvwG2DSI/edit → ID is 1OsXTACsAcf0T6prPSg4DKoJOOw_JzEMd-olLvwG2DSI). Set Up Make.com Scenario: Import the provided JSON configuration file (Integration Google Sheets, Gmail.blueprint (1).json) into Make.com. Replace the spreadsheet ID in the configuration with your own Google Sheet's ID. Update the email addresses in the "Send an Email" module if different recipients are needed. Connect your Google account in Make.com: For Google Sheets: Use devisrinivaskandikattu03@gmail.com (or your own account). For Gmail: Use 21kb1a0566@nbkrist.org (or your own restricted Gmail account). Test the Workflow: Run a test in Make.com to ensure the workflow: Finds a "Pending" row. Updates the status to "Sent". Sends the email with the correct question and answer. Check the email formatting and recipient list. Scheduling In Make.com, set the scenario to run daily at 6:00 AM. Use the scheduling settings in the Make.com interface (typically under the scenario's "Schedule" tab). Verify the time zone to ensure it aligns with your intended delivery time.

Usage Add Questions to the Sheet: Populate the "JavaQuest" Google Sheet with questions and answers. Example

A (Question)	**B (Answer)**	**C (Status)**
What is encapsulation in Java?	Encapsulation is...	Pending
What is an interface in Java?	An interface is...	Pending
Mark Status:		
Set the Status column to "Pending" for any questio want the automation to process.		
Automation in Action:		
Every day at 6:00 AM, the workflow will:		
Find the first row with "Pending" status.		
Send an email with the question and answer.		

Update the row's status to "Sent"		

Files Integration Google Sheets, Gmail.blueprint (1).json: This is the Make.com configuration file that defines the workflow. It includes the modules for searching rows, updating the sheet, and sending emails. To use it, import it into Make.com and customize the spreadsheet ID and email addresses.

How to Contribute This is an open-source project, and contributions are welcome! Here's how you can contribute:

Fork the Repository: Fork the project's GitHub repository (not provided in the description, so you'd need to create one if sharing this project publicly). Make Improvements: Add new features, such as: Supporting multiple questions per email. Customizing the email template (e.g., adding styling or logos). Adding error handling (e.g., alerting if no "Pending" rows are found). Integrating with other platforms (e.g., Slack or Microsoft Teams). Optimize the workflow for performance or scalability. Submit Pull Requests: Create a pull request with a clear description of your changes. Ensure your code or configuration is well-documented.

Suggestions: Share ideas for enhancements, such as: Adding a dashboard to monitor sent questions. Supporting multiple languages or question categories. Allowing dynamic recipient lists based on sheet data.

License

The project is licensed under the MIT License, which means:

You're free to use, modify, and distribute the code. The project comes with no warranties, and contributors are not liable for issues arising from its use. Check the full MIT License text for details if you're redistributing the project.

Contact Email: Reach out to devisrinivaskandikattu03@gmail.com for questions, feedback, or collaboration opportunities.

Linkedin: https://www.linkedin.com/in/devisrinivaskandikattu/

Potential Enhancements To make this project even more robust, consider the following ideas:

Error Handling: Add a module to handle cases where no "Pending" rows are found (e.g., send a notification to the admin). Notify if the email fails to send due to authentication or quota issues. Dynamic Recipients: Store recipient emails in the Google Sheet and dynamically pull them into the email module. Allow users to subscribe/unsubscribe via a form integrated with the sheet. Custom Email Templates: Enhance the HTML email with better formatting, such as tables, colors, or a branded header. Include additional data, like a question ID or category, in the email body. Multiple Questions: Modify

the workflow to send multiple “Pending” questions in a single email. Add logic to limit the number of questions sent per day. Analytics: Track which questions have been sent and when. Create a summary sheet or dashboard to monitor the automation’s activity. Multi-Platform Support: Integrate with other platforms like Slack, Discord, or WhatsApp for broader distribution. Localization: Support multiple languages for questions and answers. Adjust the email schedule based on recipients’ time zones.

Example Scenario Imagine you’re running a coding bootcamp and want to send daily Java interview questions to students. You:

Create a Google Sheet with 100 Java questions, each marked “Pending”. Set up the Make.com workflow using the provided JSON. Schedule it to run at 6:00 AM daily. Each morning, students receive an email like:

Subject: Javaquest Daily Interview Question

Question

What is encapsulation in Java?

Answer

Encapsulation is the bundling of data and methods that operate on that data within a single unit, typically a class...

The sheet updates to mark the question as “Sent”, and the process repeats the next day.

Conclusion:

This project is a great starting point for anyone looking to automate repetitive tasks using Google Workspace and Make.com. It’s simple yet powerful, with plenty of room for customization. If you have specific questions or need help implementing this, let me know!