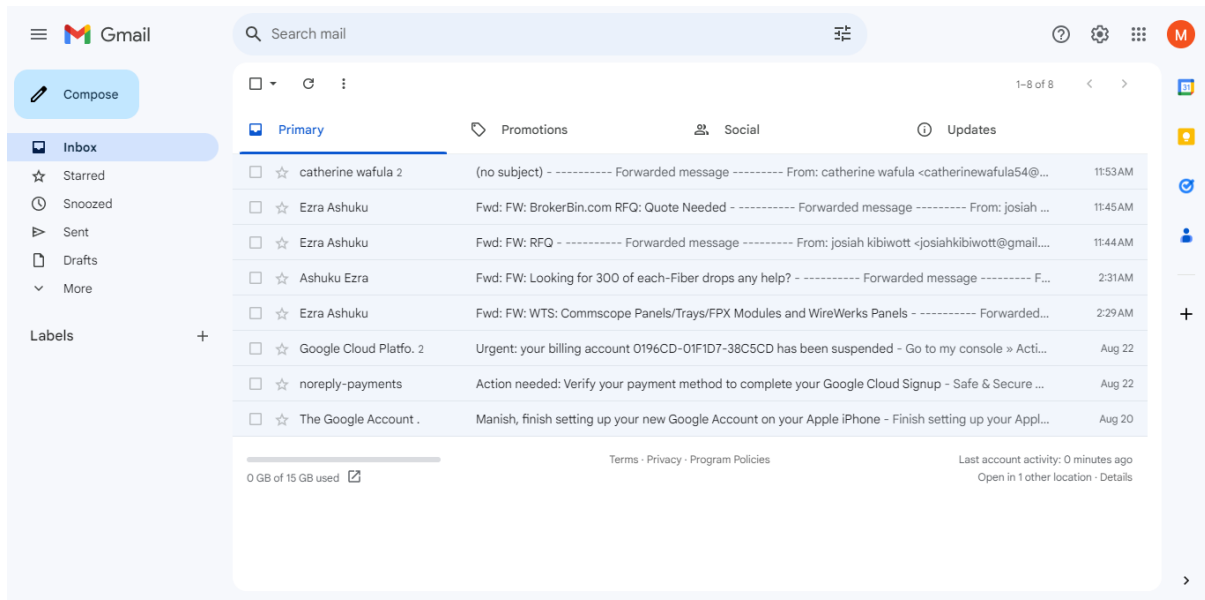


Email parser workflow and description

Input

For this particular workflow the input is an email straight from the inbox, the following image shows the received emails. However, since the emails are from different customers the structure of the email is bound to be differ.



This is a useful solution for a seller who receives a load of messages from multiple customers. Since the emails come in in unstructured format it makes it more challenging to collect, process, and analyse using traditional data tools. Unlike structured data, which is organized in rows and columns (like in a database or spreadsheet), unstructured data comes in various unpredictable forms.

Output

Output is a spread sheet containing the customer email address, products inquired and date the email was received. Upon successful running of the workflow a raw is appended to a google sheet containing the relevant information required to respond to a customer's query. This is very efficient as the data is in a structured format and can be integrated in customer support systems for efficient and timely handling of customers query's to a business

A screenshot of a Google Sheet titled 'products-enquiry'. The sheet contains a table with 5 columns: A1, A, B, C, and D. The data is as follows:

A1	A	B	C	D
1	joshiah kibiwott	2024-08-13T17:1	Commscope-FMT fiber Storage 1RU 19" Black sliding tray, Commscope-FPX MPO-LCA Module Left - 24 port, Commscope-FPX MPO-LCA Module Right - 24 port, Commscope-FPX MPC	
2	ashukuezra207	2024-08-24T23:	CORNING 37R44449748-400F SC/APC ROC BX8, CORNING 37R44449748-500F SC/APC ROC BX6	
3	ezrashuku@gmail	2024-08-25T08:	SA - SA2-HGD-750-40/51-10A (Need both Amp and Housing), 539321, 544458	
4	ezrashuku@gmail	2024-08-25T08:	UPL-NM-12 JMA WIRELESS NEW 12 14	
5	catherinewafula	2024-08-25T08:	Iphone 15 pro, samsung galaxy s24, redmi 13, xiaomi redmi 13c, MacBook pro	
6				

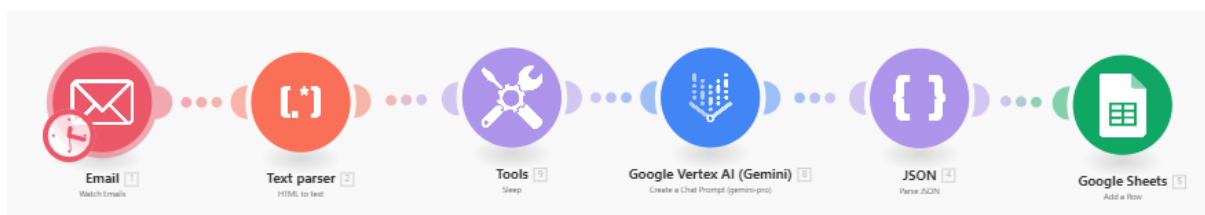
Description

The workflow is enabled on make.com formerly known as integromat. To automate tasks in Make.com, follow these steps:

1. **Creating a Scenario:** You start by creating a new scenario, which is a workflow that connects different apps and services. For example, email, LLMs like ChatGPT, google sheets etcetera.
2. **Add Modules:** Add the apps and services you want to connect as modules. Each module represents a specific action (e.g., sending an email, retrieving data). For instance, retrieving data from emails the module would be a text parser.
3. **Set Triggers:** Choose a trigger module that starts the scenario when a certain event occurs (e.g., a new email arrives, a file is updated). For instance, when a new email arrives the email is retrieved from the inbox to parse the data in it using NLP.
4. **Define Actions:** Add modules that perform actions based on the trigger (e.g., send a notification, create a record in a database). For instance, once the appropriate data is retrieved from an email the data is append onto a spread sheet.
5. **Set Filters and Conditions:** Apply filters or conditions to control when actions should be executed. This is especially useful when you wish to abide by a particular applications rate limit. For instance, to avoid error with code 429. We add a delay module called sleep to control the waiting time before sending a request to a specific API e.g., Gemini or ChatGPT.
6. **Test and Run:** Test the scenario to ensure it works as expected, then activate it to run automatically based on your trigger conditions.

For our specific context this is how the workflow is structured. We have two different workflows on that uses Gemini for natural language processing and another workflow that sends a Post request to an application that can potentially serve Llama an open-source large language model.

Below is a picture showcasing a workflow that utilizes Gemini for natural language processing.

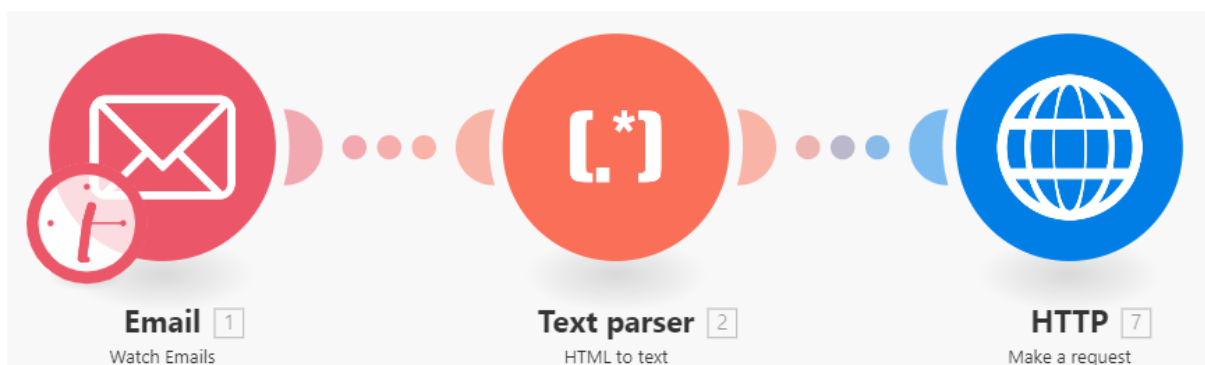


The scenario has 5 modules each dependent on the previous modules output apart from email. A description of what each module does is as follows;

- **Email** – Watch email. This module watches incoming mails in the Inbox of the connected email. The module queries for unread emails at specified intervals. If it finds an email it marks the email as read and passes the HTML content of the email to the Text-parser. This is the starting point of transforming unstructured data into structured data containing rows and columns
- **Text Parser** – HTML to Text. This is a custom text parser used to convert HTML to its corresponding text. For this case the text-parser takes the body of the email with all its HTML tags then transforms it to text by removing the HTML tags. The text is required for easier prompting of NLP models to get information within the parsed text.

- **Tools – Sleep.** App modules return the `RateLimitError` when you reach the API rate limit for the app. The rate limit is the number of requests you can send to an app over a period of time, typically over a second or a minute. The `RateLimitError` corresponds to the HTTP error code 429. When you reach the app rate limit, the app will block further requests until the limiting time period passes. To avoid this the sleep module is used to delay the request for a specified period of time so that the workflow runs uninterrupted.
- **Google vertex ai (Gemini)** – Create a Chat Prompt. With Google Vertex AI (Gemini) modules in Make, you can create text prompts (text-bison), chat prompts (chat-bison and Gemini-pro), and analyse images and video. This acts as an NLP used to extract information from the email body. A prompt is constructed containing the email body, sender's email and the date of receiving the email. In addition, the prompt request that after extracting data from the email body return JSON and only JSON containing only the sender, products, and date. This JSON is then parsed to the next module JSON.
- **JSON – Parse JSON.** This module is used to parse the received JSON response from Gemini. Put it in a format that google sheets can understand
- **Google sheets – Add row.** This module is used to add rows to an existing spread sheet in a google drive. The customer information, product information and date are appended to the sheet on each successful run of the workflow.

Lastly below is a picture showcasing a workflow that could potentially utilize Llama 3.1 an open-source large language model.



The workflow has only 3 modules. The only difference with the previous workflow is the HTTP module. This module is used to make a Post request to a FastAPI application which could serve the Llama 3.1.

After receiving the post request containing the parsed email body, sender's email and date. The email body is used to prompt Llama 3.1 to extract the product information being inquired for then the information is saved in a spread sheet by using python packages like Pandas.

This gives more flexibility as one can setup rules if there is no product information in the email the email is discarded.