Make.com: Scenarios, Routers and AI Agents

Make is a visual automation tool. You connect apps as boxes on a canvas and draw the flow between them. This guide takes you from your first scenario to the Resume Analysis Agent from the class. It uses fake data and a test account throughout.

1. Words you need first

Word	Meaning
Scenario	One automation: a chain of modules that runs in order.
Module	One step, such as "Watch responses" or "Add a row".
Trigger	The first module, which starts the scenario (a new form response, a schedule, a webhook).
Bundle	One item of data passing between modules.
Router	A split that sends each bundle down one or more paths based on filters.
Filter	A condition on a path, for example "score is above 70".
Connection	The saved login to an app (a credential). Use a test account.
Operation	Spend of the plan's credits per module run. Check the plan page for current credit rules.

2. The Make screen

The left menu has: **Org**, **Scenarios**, **AI Agents**, **Credentials**, **Webhooks**, **MCP Toolboxes**, **Templates**, **Data stores** and **More**. The canvas in the middle is where you build. The bottom toolbar has **Run once**, a schedule toggle (for example every 15 minutes) and the undo and settings icons.

3. Your first scenario: form to sheet

1. Click **Scenarios**, then **Create a new scenario**.
2. Add a **Google Forms** module, choose **Watch Responses**, and connect the test account. Pick a test form with two fields.

3. Add a **Google Sheets** module and choose **Add a Row**. Select a test sheet.
4. Map each form answer to a column using the mapping panel (click a field, choose the answer).
5. Click **Run once**, then submit one test response to the form.

You should see: the module bubbles turn green, the sheet gets a new row, and the output panel shows the values you mapped.

4. Scheduling and settings

- **Schedule:** the toggle sets how often the scenario checks for new items. The class example was every 15 minutes. Shorter intervals use more operations.
- **Sequential running:** keep it on when order matters, so two runs cannot write at the same time.
- **Error handling:** right-click a module to add an error route. Route failures to a sheet row or a message so they are visible.
- **Notifications:** decide when Make emails you about failed runs. Turn on notifications for failures, not for every success.

> **Before you enable a scenario,** check three things: its operations per month against your plan, its notification setting, and what happens on failure. A scenario left running on a schedule keeps spending operations.

5. Routers and filters

A router is a fork in the road. The class example, the Resume Analysis Agent, uses one.

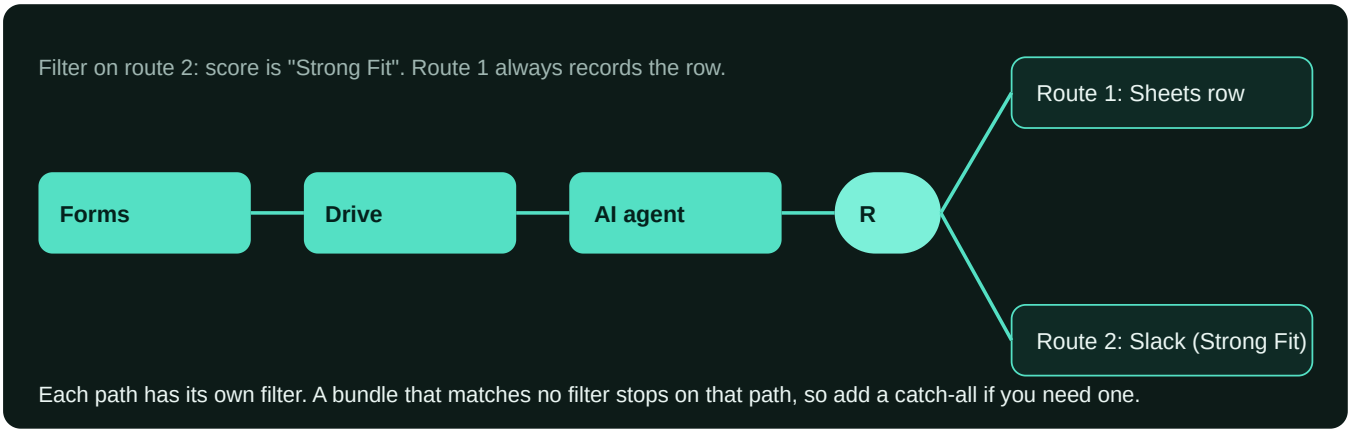


Figure 1. A router splits bundles into paths. Each path has a filter.

1. Click the wrench icon between two modules and choose **Add router**.
2. Add a filter to each route (for example the AI score field equals "Strong Fit").

3. Test with one bundle that matches and one that does not. Confirm only the right route runs.

6. The Resume Analysis Agent (class example)

The class scenario was built to read a candidate's resume and sort the candidate. The full chain is:

1. **Google Forms (Watch Responses):** the candidate submits a form with a resume file link.
2. **Google Drive (Download a File):** the resume file is fetched from Drive.
3. **AI agent:** reads the resume, uses a knowledge source (the job criteria) and returns a fit rating.
4. **Router:** route 1 adds a row to Google Sheets for every candidate; route 2 sends a Slack message only for "Strong Fit" candidates.

The agent in the screenshot also had an **Add tool** option, an **Add MCP** option, a **Chat** option and a knowledge attachment. The AI model module was an OpenAI connection in the class.

> **Privacy.** A resume contains personal data. For practice, use a fake resume with a made-up name and email. Do not send real candidates' files to a test scenario, and do not post Slack messages to a shared channel during testing.

Build checklist

Step	You should see
Forms trigger runs on a test response	One bundle with the form fields
Drive download returns a file	A file name and size in the output
AI step returns a fixed-format rating	One word from your allowed list (for example Strong Fit, Possible, No Fit)
Router sends only matching bundles to Slack	Slack message only for Strong Fit
Failed AI step	Error route writes a row marked "needs attention"

7. Maia: building with a description

Maia is Make's AI assistant (labelled Beta in the class screenshots). You describe what you want in plain English, and Maia proposes modules. The panel offers "Continue where we left off", "Get inspiration" and "Ask a question about Make".

1. Open a scenario and click **Build with Maia**.

2. Describe the goal in one sentence: "When a form is submitted, download the attached file, ask an AI model to rate it, and add a row to a sheet."
3. Review every module Maia proposes. Check each connection, field mapping and filter yourself.
4. Never trust a generated scenario without running it on fake data first.

Maia is a drafting aid. The person who runs the scenario owns its results, so review every step.

8. MCP Toolboxes and AI Agents in Make

MCP (Model Context Protocol) is a standard way for an AI app to use outside tools. Make's MCP Toolboxes group tools that an agent can call. Give an agent only the tools it needs, and read-only tools where possible. The same least-permission rule from the skills and agents guide applies here.

9. Failure, retries and cost

Issue	What to do
Module fails on a bad file	Route the error to a "needs attention" row; fix the input; re-run that bundle.
Scenario runs too often	Lengthen the schedule or use a webhook trigger instead of polling.
Operations run out	Check the plan usage page and reduce the schedule or the number of modules per run.
Duplicate rows	Store the form response ID and skip responses already in the sheet.

Check Make's current plan and credit page before you estimate cost. Plans and credit rules change.

10. Quick check

1. What starts a scenario?
2. What does a filter do on a router path?
3. Why should you use a test connection?
4. What should you do before enabling a scenario that Maia built?

Answers: 1. The trigger module (for example a new form response). 2. Decides whether a bundle goes down that path. 3. So a mistake cannot touch real accounts or data. 4. Run it on fake data and check every module and filter.

11. Glossary and further reading

- Make help centre: help.make.com (scenarios, routers, Maia and MCP pages).
- Make pricing and credit page: check the live page before you estimate operations.