

Example 1 — HTTP quickstart

The simplest way to call Ask Sage from Power Automate: a manually-triggered flow that takes a question, sends it to the Ask Sage API with the built-in **HTTP** action, and shows the answer.

Best for: testing, prototypes, one-off flows.

Requires: a Power Automate plan with premium connectors (the HTTP action is premium). See the Prerequisites section of the Ask Sage Power Automate docs.

Two ways to get this flow: import the pre-built solution and open the **Initial HTTP Connection Template** flow — then follow the post-import steps to set your API key (the Importing a solution section of the Ask Sage Power Automate docs) — or build it by hand with the steps below.

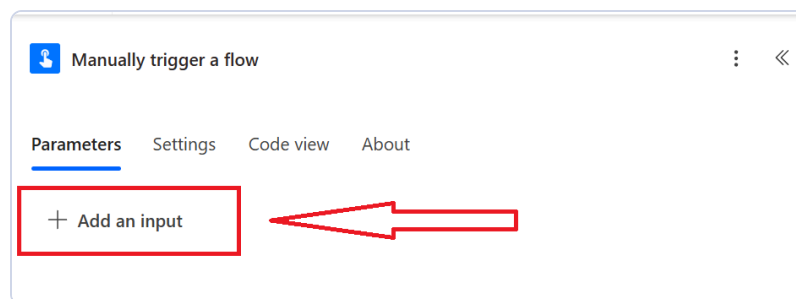
Build it by hand

1. Create the flow

1. **+ Create** → **Instant cloud flow**.
2. Name it `Ask Sage HTTP Test`, choose **Manually trigger a flow**, click **Create**.

2. Add a question input

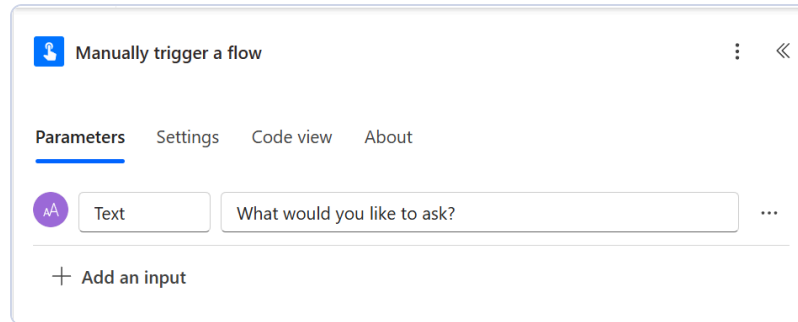
1. Click the **Manually trigger a flow** trigger → **+ Add an input**.



2. Choose **Text** and configure:

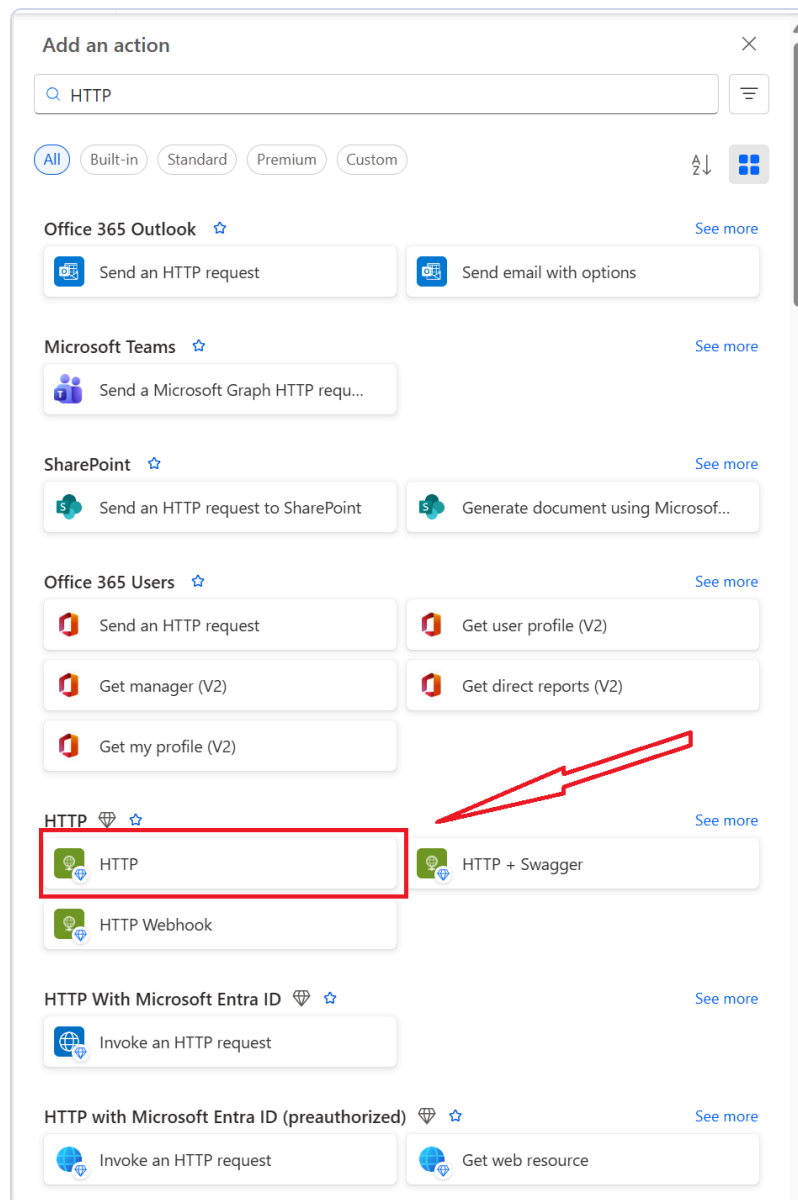
- **Input:** `Question`

- **Please enter your input:** What would you like to ask?



3. Add the HTTP action

1. + New step → search HTTP → select HTTP.



2. Configure the request:

Field	Value
Method	POST
URI	https://api.asksage.ai/server/query
Headers	x-access-tokens : YOUR_ASK_SAGE_API_KEY and Content-Type : application/json
Body	{ "message": "" }

3. Put the cursor inside the quotes after "message": , click the **lightning bolt**, and pick **Question** (your trigger input) from dynamic content.

The screenshot shows an HTTP client interface with the following configuration:

- URI:** https://api.asksage.ai/server/query
- Method:** POST
- Headers:**
 - Key: x-access-tokens, Value: access_token
 - Key: Content-Type, Value: application/json
 - Buttons: Enter key, Enter value
- Queries:**
 - Buttons: Enter key, Enter value
- Body:**

```
{
  "message": ""
}
```
- Cookie:** Enter HTTP cookie
- Advanced parameters:** Showing 0 of 1, Show all, Clear all

The screenshot shows an HTTP client interface with the following configuration:

- URI: `https://api.asksage.ai/server/query`
- Method: `POST`
- Headers:
 - `x-access-tokens`: `access_token`
 - `Content-Type`: `application/json`
- Queries: (Empty)
- Body:

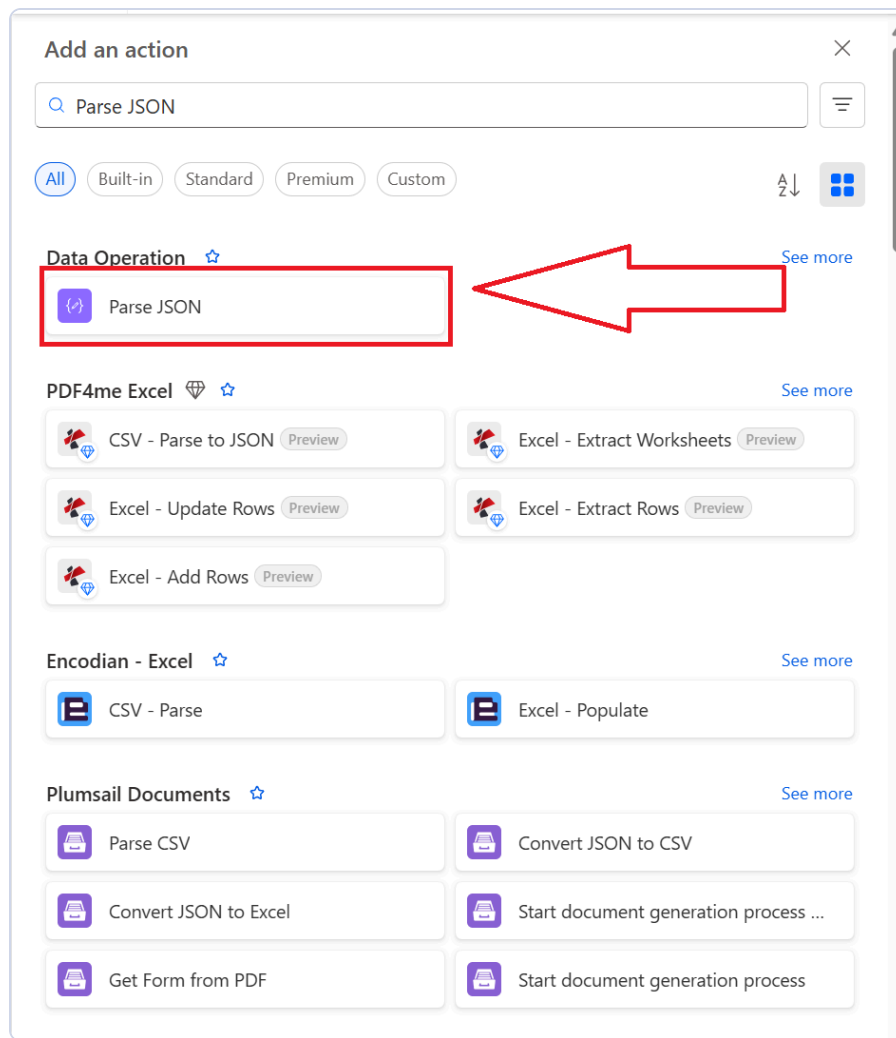
```
{
  "message": ""
}
```
- Cookie: (Empty)
- Advanced parameters: Showing 0 of 1

The response preview on the right shows a list of fields. A red arrow points to the **Text** field, which contains the value `What would you like to ask?`.

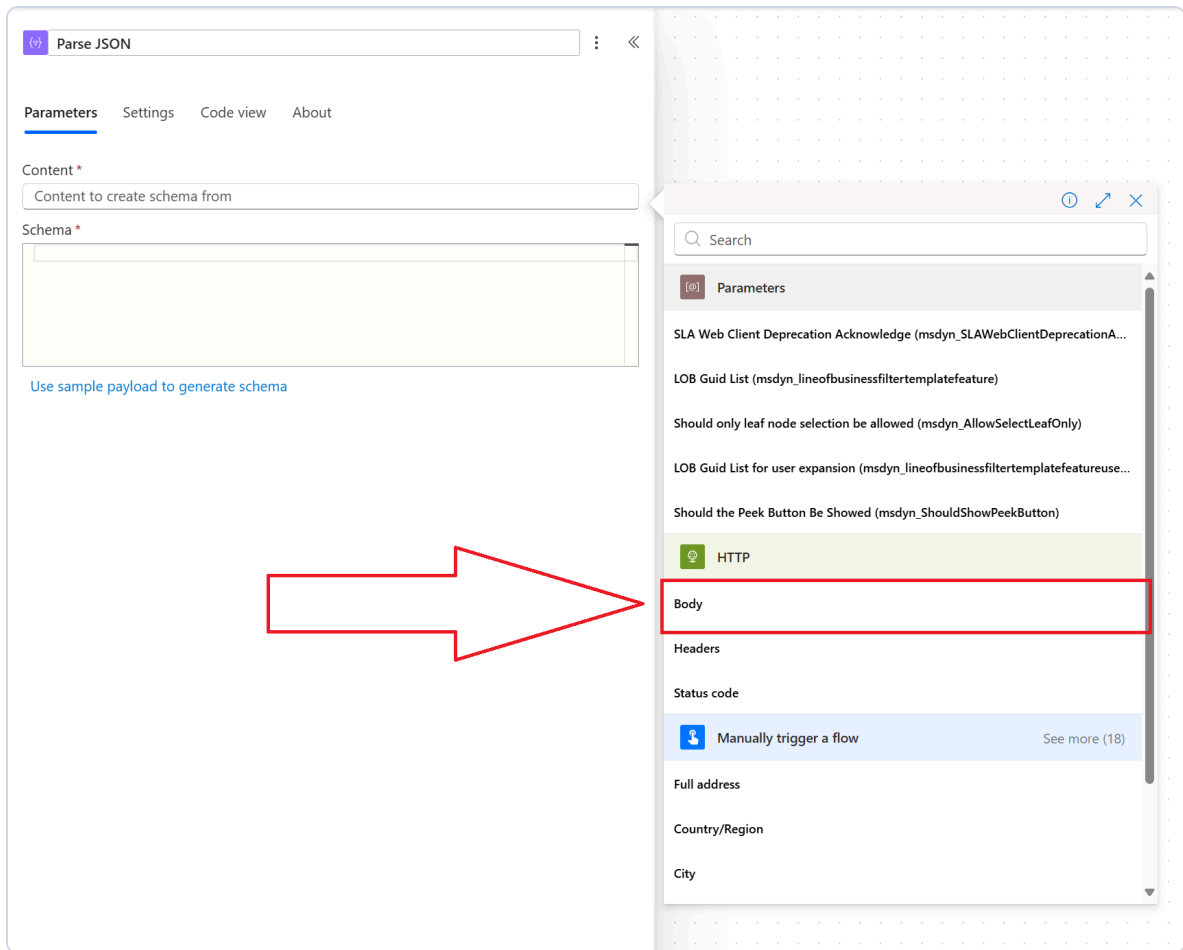
Replace `YOUR_ASK_SAGE_API_KEY` with your real key — and for anything beyond a quick test, store it securely instead of pasting it here. See the Authentication section of the Ask Sage Power Automate docs.

4. Parse the response

1. **+ New step** → search **Parse JSON** → select **Parse JSON**.



2. **Content:** click the lightning bolt → select **Body** from the HTTP action.



3. **Schema:** click **Generate from sample** and paste a sample response, then **Done**:

```
{  
  "message": "Sample response from Ask Sage",  
  "usage": { "prompt_tokens": 10, "completion_tokens": 20, "total_tokens": 30 }  
}
```

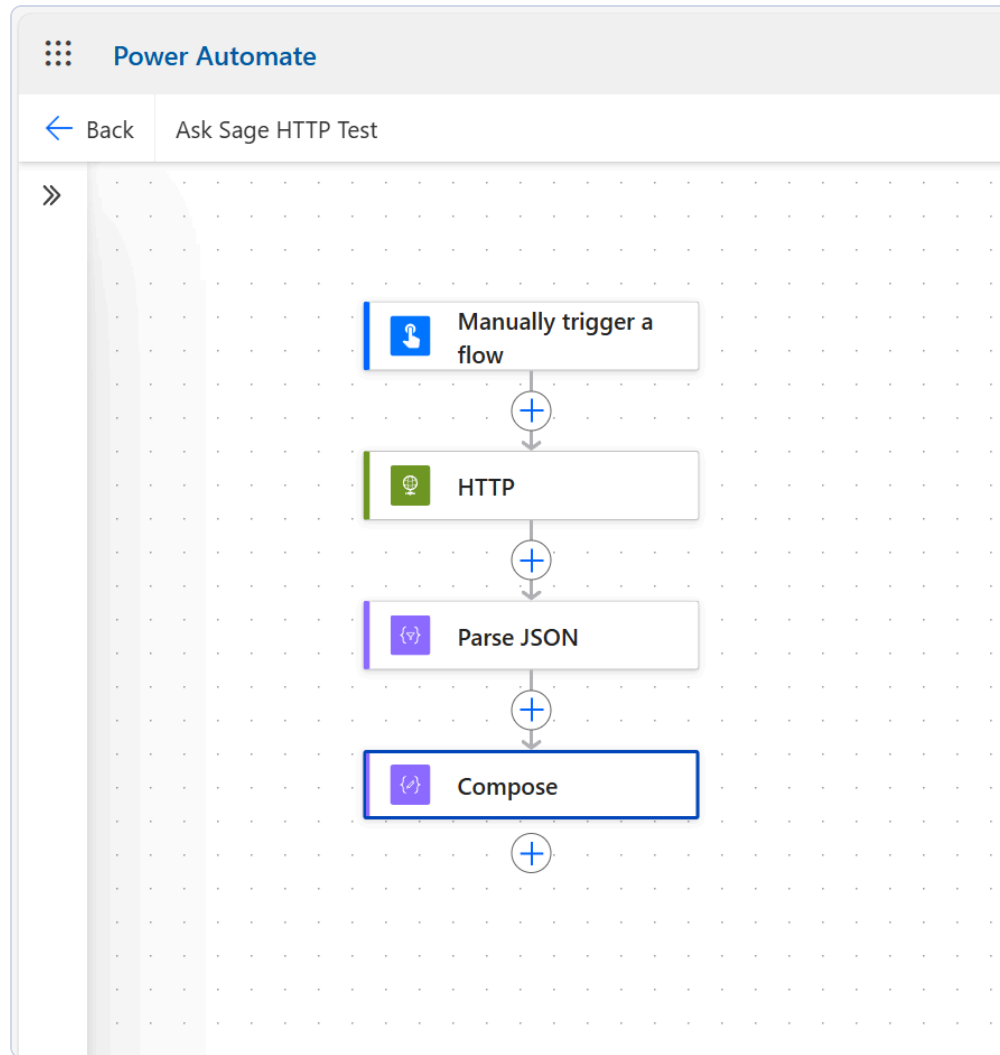
5. Show the answer

1. **+ New step** → **Compose**.
2. **Inputs:** lightning bolt → select **message** from Parse JSON.

6. Save and test

1. **Save**, then **Test** → **Manually** → **Test**.
2. Enter a question (e.g. *What is artificial intelligence?*), **Run flow** → **Done**.
3. Open the **Compose** action in the run to see the answer.

Full flow



Troubleshooting

- **401 / “token is invalid”**: confirm the header is `x-access-tokens` (not `Authorization`), and that the value is your API key with no stray spaces.
- **HTTP action missing**: it’s a premium connector — check your license.
- **Empty message**: open the run history and inspect the raw HTTP **Body**; re-generate the Parse JSON schema from that actual response.

See also: the Authentication section of the Ask Sage Power Automate docs ·
the Prerequisites section of the Ask Sage Power Automate docs ·
custom connector example.

