Alchemer Workflow Custom Events Initiator

Overview

A **webhook** is an HTTP-based callback that allows external systems to send data to Alchemer in real time when an event occurs. Organizations use webhooks to trigger automated processes as soon as data is available, without polling or manual intervention.

The Alchemer Webhook Workflow Initiator allows Alchemer to start a workflow run when an HTTP POST or GET request is received at a generated webhook URL. Depending on the action selected, a single workflow run is triggered per request, or one workflow run is triggered for each object in an array — fully automating event-driven processing and downstream integrations.

Common uses for the Alchemer Webhook initiator

- Trigger a workflow instantly when an external system sends a webhook
- Process each object in a webhook array as an independent workflow run
- Pass webhook payload data into workflow steps as merge codes
- Secure incoming webhooks with custom headers, IP whitelisting, or basic authentication
- Automate event-driven workflows without manual data entry or scheduled polling

What can the Alchemer Webhook initiator do?

- Webhook listener (single)
- Webhook listener (multiple)

You will need

- An external system capable of sending HTTP POST or GET requests
- An Alchemer plan that includes integrations with Workflow and the Integration Manager permission enabled.
  - Contact us if you are unsure if your plan includes integrations.

Setup Alchemer Webhook initiator in Workflow

Webhook | Webhook listener (single)

Use this action to start a single workflow run each time a webhook request is received. Each request triggers one workflow execution, with the payload fields available as merge codes in downstream steps.

You will need:

- An external system configured to send HTTP POST or GET requests to the Alchemer webhook URL
- The custom field names (keys) your webhook payload will include

Configure the action

1. Open your workflow in **Workflow Builder**.
2. On the **Select Initiator** pop up, select the **Webhook** initiator.
3. Select **Webhook | Webhook listener (single)**.
4. **Webhook listener | Settings — Webhook URL:** Copy the generated webhook URL. Configure your external system to send POST or GET requests to this endpoint to trigger the workflow.
5. **Webhook listener | Settings — Custom Fields:** Add the custom fields your webhook payload will include. These fields define what data the webhook should accept and will be available as merge codes in the workflow. Fields can be passed as query parameters or as JSON in the body of a POST request.
6. **Webhook listener | Settings — Custom Headers (optional):** Add one or more HTTP header key/value pairs to secure your webhook. Incoming requests must include these headers to be accepted.
7. **Webhook listener | Settings — IP Whitelist (optional):** Enter the IP address or addresses that requests will originate from. Incoming requests must match one of these addresses to be accepted.
8. **Webhook listener | Settings — Basic Authentication (optional):** Enter a username and password to require basic authentication on incoming webhook requests.
9. Save the action.

Status codes

- 200: Webhook received and workflow successfully triggered
- 400: The request was rejected due to an authentication or validation error

Webhook | Webhook listener (multiple)

Use this action when your webhook payload contains an array of objects. Each object in the array triggers a separate, independent workflow run — with the key/value pairs from that object available as merge codes in the workflow.

You will need:

- An external system configured to send HTTP POST or GET requests containing an array of objects to the Alchemer webhook URL
- The custom field names (keys) each object in the array will include

Configure the action

1. Open your workflow in **Workflow Builder**.

2. On the **Select Initiator** pop up, select the **Webhook** initiator.

3. Select **Webhook | Webhook listener (multiple)**.

4. **Webhook listener | Settings — Webhook URL:** Copy the generated webhook URL. Configure your external system to send POST or GET requests to this endpoint. Each object in the array sent to this URL will trigger a separate workflow run.

5. **Webhook listener | Settings — Custom Fields:** Add the custom fields that each object in your webhook array will include. These fields define what data the webhook should accept and will be available as merge codes for each individual workflow run. Fields can be passed as query parameters or as JSON in the body of a POST request.

6. **Webhook listener | Settings — Custom Headers (optional):** Add one or more HTTP header key/value pairs to secure your webhook. Incoming requests must include these headers to be accepted.

7. **Webhook listener | Settings — IP Whitelist (optional):** Enter the IP address or addresses that requests will originate from. Incoming requests must match one of these addresses to be accepted.

8. **Webhook listener | Settings — Basic Authentication (optional):** Enter a username and password to require basic authentication on incoming webhook requests.

9. Save the action.

Status codes

- 200: Webhook received and a separate workflow run successfully triggered for each object in the array
- 400: The request was rejected due to an authentication or validation error

Testing and Troubleshooting

Testing and Validation

How to test

- Send a test HTTP POST or GET request to the generated webhook URL from your external system or a tool such as Postman.
  - For the (multiple) action, send a request containing an array of objects and confirm that a separate workflow run is created for each object.
- Monitor individual runs in the **Monitor** tab.
  - Click on each run to view the inputs received and metadata outputs.

How to verify results

- Set a send email action to yourself with action metadata included.
- Confirm that the expected payload fields are available as merge codes and contain the correct values.

Monitoring Integration Activity

Where to find logs

- Go to **Monitor**.
- Check the individual workflow steps.

What logs display

- Inputs received per webhook request or per object in the array
- Workflow output or returned status

Troubleshooting

Authentication issues

- Custom header key/value pairs do not match what the sending system is passing
- Sending IP address not included in the IP whitelist
- Incorrect basic authentication username or password

Webhook issues

- Custom field names do not match the keys in the incoming payload
- Payload is not valid JSON or is malformed
- Array not correctly structured for the (multiple) action

API errors

- Request validation failures
- Endpoint or runtime errors

FAQs

What permissions do I need?

Integration Manager in Alchemer.

When does the integration run?

In real time, as soon as a webhook request is received at the generated URL.

What is the difference between (single) and (multiple)?

The (single) action triggers one workflow run per incoming request. The (multiple) action triggers a separate workflow run for each object in an array passed in the request payload.

Why isn’t my workflow triggering?

Check the Monitor tab for authentication errors, payload formatting issues, or IP whitelist mismatches.

What if I need additional functionality?

Contact Alchemer Support for enhancement requests.

Related Articles