

Zapier Webhooks: How to Use Them as Trigger and Action

Zapier's **Webhooks** feature allows you to connect apps that don't have native Zapier integrations or to send/receive custom data between systems. Webhooks are HTTP requests that carry data between applications in real time.

1. What is a Webhook?

A webhook is a way for one application to send data to another application as soon as an event occurs. Unlike APIs that require polling, webhooks push data instantly.

2. Why Use Zapier Webhooks?

- Integrate apps without native Zapier support.
 - Send custom payloads to external systems.
 - Receive data from external systems to trigger workflows.
 - Enable advanced automation scenarios.
-

3. Webhooks in Zapier

Zapier supports webhooks in two ways:

- **Trigger:** Zap starts when Zapier receives data via a webhook.
 - **Action:** Zap sends data to a webhook URL.
-

4. Using Webhook as a Trigger

Step-by-Step Guide

1. **Create a New Zap**
 - Go to Zapier Dashboard.
 - Click **Create Zap**.
2. **Choose Trigger App**
 - Search for **Webhooks by Zapier**.
 - Select **Catch Hook** (most common option).
3. **Copy the Webhook URL**

- Zapier will generate a unique URL.
- Example: <https://hooks.zapier.com/hooks/catch/123456/abcdef>

4. Send Data to the Webhook

- From your external app/system, configure it to send an HTTP POST request to the Zapier webhook URL.
- Include the data in JSON or form-data format.

5. Example Payload:

6. Test the Trigger

- Zapier will wait for a sample request.
- Send a test request from your app.
- Zapier will display the received data.

7. Use Data in Next Steps

- Map the webhook fields to actions in subsequent steps (e.g., send email, update CRM).
-

5. Using Webhook as an Action

Step-by-Step Guide

1. Add Action to Your Zap

- After setting up a trigger, click **Add Action**.
- Search for **Webhooks by Zapier**.

2. Choose Action Event

- Options:
 - **POST**: Send data to a URL.
 - **GET**: Retrieve data from a URL.
 - **PUT, PATCH, DELETE**: For advanced API operations.

3. Configure the Webhook

- Enter the target URL (where you want to send data).
- Choose the method (POST is most common).
- Add **Payload Type**:
 - JSON (recommended)
 - Form
- Map fields from previous steps to the payload.

4. Example POST Configuration:

- URL: <https://api.example.com/orders>
- Payload:

5. Headers (Optional)

- Add authentication or content-type headers if required.
- Example:

6. Test the Action

- Zapier will send a test request.
 - Verify response from the target system.
-

6. Best Practices

- **Security:** Use authentication tokens or secret keys for webhook URLs.
 - **Validation:** Ensure payload structure matches the receiving system's requirements.
 - **Error Handling:** Check Zapier task history for failed requests.
 - **Documentation:** Keep a record of webhook URLs and payload formats.
-

7. Common Use Cases

- Receive form submissions from a custom website.
 - Send data to a custom API after an e-commerce order.
 - Connect IoT devices to Zapier workflows.
 - Automate notifications when external systems update.
-

Summary

- **Trigger:** Use “Catch Hook” to start a Zap when data is sent to Zapier.
 - **Action:** Use “POST” (or other methods) to send data from Zapier to an external system.
 - Webhooks enable flexible, real-time integrations beyond standard app connectors.
-