

Valerie Wade Writing Sample
July 2021
<https://vcwade.com/tech>

Introduction to Webhooks

Webhooks let you respond to events in real-time. For example, a vendor may want to know exactly when someone places an order on their store, when one of their shipments gets delivered, or whenever any other package-tracking event occurs.

Think of webhooks as a way to receive automatic updates about something that has happened. These events can be as simple as receiving an update about the status of an order. Some companies call them push notifications. Consider tracking a package without a webhook. You would likely need to do the following:

- Write code to call servers every few minutes.
- Integrate your code for each of your shipments.
- Stack up countless API calls to receive 'no update.'
- Waste server time and bandwidth.

Webhooks help address this inefficiency. Here's how:

- Create a webhook in your admin panel.
- When an event is updated by the carrier, your server receives an alert about the update in real time.
- The result is one HTTP request rather than many.

Webhooks vs APIs

How are APIs different from webhooks? APIs help you get the information you requested. Webhooks push information to you automatically once they are set up in your admin panel.

Types of Webhooks

Only one webhook can exist for each event. Here are some examples of webhooks you might use as a vendor:

Event	Description
Tracking	Updates about your shipment status.
Batch completed	A collection of new shipping labels is complete.
Shipment rate updated	Updates shipping rate.
Carrier connected	Carrier is connected to your account.
Sales orders imported	Sales orders have been imported to your account.
Store refresh completed	Your store data has been refreshed.

HTTPS Endpoint URLs

You need your own HTTPS endpoint URL to help process your webhook alerts. From here on out, we'll refer to the HTTPS endpoint URL as your 'endpoint.' If you don't have your own servers, we recommend getting a free endpoint at <https://webhook.site> `target="_blank">webhook.site`.

For those users who have their own servers, here's what you need to set-up your webhooks:

1. A web server. Preferably HTTPS for security reasons.
2. Your server must accept the JSON request format.
3. Your server must have a publicly-accessible URL so your admin panel can access it.

****Warning:****

Since your endpoint will be public on the Internet, take all of the normal security precautions necessary for any public web server to guard against malicious attacks.

Configuring Webhooks with the API

Below are examples of vendor webhooks along with their required values:

Event	Value
Batch completed	`batch`
Shipment rate updated	`rate`
Any tracking event	`track`
Carrier connected	`carrier_connected`
Sales orders imported	`sales_orders_imported`
Order source refresh complete	`order_source_refresh_complete`

****Testing a Webhook****

In the following batch completed example call, replace `https://yourserver.com` with your new endpoint URL. We'll use `https://yourserver.com` for the following examples.

****Request****

```
curl -iX POST https://api.yourstore.com/v1/environment/webhooks \
-H 'Content-Type: application/json' \
-H 'API-Key: __YOUR_API_KEY_HERE__' \
-d '{
"url": "https://your-server.com",
"event": "batch"
}'
```

****JSON Response****

```
{
"webhook_id": "5519",
"url": "https://your-server.com",
"event": "batch"
}
```

****Warning: HTTP 409 Conflict****

If a webhook already exists in your account, you'll receive an `HTTP 409 Conflict` error after your request is sent. If this occurs, you need to review your active webhooks.

Tracking Status Codes

Below are examples of tracking status codes and descriptions:

Status Code	Description
AC	The package was accepted by the carrier.
IT	The package is in transit to its destination.
DE	Hooray, the package was delivered.
EX	An exception was identified.
UN	The tracking status is unknown.
AT	An attempt was made to deliver the package.