Working with Webhooks

4.7.1

What is a Webhook?

A webhook is an HTTP callback, or an HTTP POST, to a specified URL that notifies your application when a particular activity or “event” has occurred in one of your resources on the platform. The concept is simple. Think of asking someone "tell me right away if X happens". That "someone" is the webhook provider, and you are the application.

Webhooks enable applications to get real-time data, because they are triggered by particular activities or events. With webhooks, applications are more efficient because they no longer need to have a polling mechanism. A polling mechanism is a way of repeatedly requesting information from the service, until a condition is met. Imagine needing to ask someone, again and again, "Has X happened yet? Has X happened yet?" Annoying, right? That's polling. Polling degrades the performance of both the client and server due to repeatedly processing requests and responses. Furthermore, polling isn't real-time, because polls happen at fixed intervals. If an event occurs right after the last time you polled, your application won't learn about the changed status until the poll interval expires and the next poll occurs.

Webhooks are also known as reverse APIs, because applications subscribe to a webhook server by registering with the webhook provider. During this registration process, the application provides a URI to be called by the server when the target activity or event occurs. This URI is typically an API on the application side that the server calls when the webhook is triggered.

When the webhook is triggered, the server sends a notification by becoming the caller and makes a request to the provided URI. This URI represents the API for the application, and the application becomes the callee and consumes the request. As a result, for webhooks, the roles are reversed; the server becomes the client and the client becomes the server. Multiple applications can subscribe to a single webhook server.

Examples:

- The Cisco DNA Center platform provides webhooks that enable third-party applications to receive network data when specified events occur. You can have your application registered with a particular REST endpoint URI that receives a message from Cisco DNAC when a particular event occurs. For example, if a network device becomes unreachable, the Cisco DNAC webhook can send an HTTP POST to the URI your app has registered on the Cisco DNAC. Your application then receives all the details of the outage in a JSON object from that HTTP POST so it can take action as appropriate. In this case, Cisco DNAC is the webhook provider.

- You can create a webhook to have Cisco Webex Teams notify you whenever new messages are posted in a particular room. This way, instead of your app making repeated calls to the Teams API to determine whether a new message has been posted, the webhook automatically notifies you of each message. In this case, Cisco Webex Teams is the webhook provider.

4.7.2

Consuming a Webhook

Consuming a webhook

In order to receive a notification from a webhook provider, the application must meet certain requirements:

- The application must be running at all times to receive HTTP POST requests.
- The application must register a URI on the webhook provider so that the provider knows where to send a notification when target events occur.

In addition to these two requirements, the application must handle the incoming notifications from the webhook server.

Because working with webhooks involves third parties, it can sometimes be challenging to ensure everything is working properly. There are many free online tools that ensure your application can receive notifications from a webhook. Many can also give you a preview of the content provided in the webhook notification. These tools can be found by searching for "webhook tester", and can be useful when designing and implementing your application.

4.6 API Rate Limits

4.8 Troubleshooting API Calls