Webhooks: The Building Blocks of an Event-Driven Architecture

23 Mar 2020 9:14am, by Jennifer Riggins



Spiders don't just go and catch food all the time, they just get a notification on the web. And when w + frozen meal in the microwave — we don't continuously check on it. We set the time and temperature and we just know when it's done.

These are the ways Typeform Developer Advocate Nicolas Grenié sums up webhooks. A webhook programmatically provides a notification when something happens on a platform or service.

Programmer Jeff Lindsay created the concept in 2007, when he was looking for the web's equivalent of what he called the "pipe" that would be created by a chain of commands. He coined the phrase "web hooks" as a name for these "user-defined callbacks made with HTTP POST." He saw this "open ended infrastructure" as ripe with possibilities for the Web and the then still-growing API ecosystem.

These days at Typeform, Grenié says, responding to an online form, for instance, will trigger a webhook to a third-party service that sends you a thank you text message and an agent will be notified you sent a support ticket. This webhook integration is what's linking Typeform to the customer relationship management system (CRM), the database, and the content management system (CMS). It helps you with the data flow and connects two systems together.

Webhooks drive services such as Zapier and IFTTT. A bedrock for event-driven architecture aims to simplify building complex user experience chains, connecting your service to other ecosystems and thousands of services.

“Webhooks make it simpler for people to integrate with your product because they know that they will receive data from you and it’s just ‘data in, data out’,” Grenié said.

Webhooks for an API Strategy

Grenié told The New Stack, over Zoom, that webhooks could leverage serverless functions like Lambda on Amazon Web Services or Google Cloud Functions.

He said that webhooks are an easy way for organizations to kick off their application programming interface (API) strategy. It allows for simpler building blocks that can later be turned into an API — or maybe that’s not even necessary.

“At Typeform, if we didn’t have webhooks before the API, we probably wouldn’t have grown the adoption of the platform,” Grenié said.

He argues you should be building a better developer experience around webhooks. He says this starts with consistency.

“About two years ago, when someone was entering a Typeform, we would send a webhook event which had an adjacent payload with the answers of that Typeform. You would answer the form again with the same input. The webhook payload would be different. The answers would be in a different order,” he said. As a result, developers could not build reliable apps with this inconsistency.

> “Having an API to manage Webhooks should be as crucial as having webhooks itself. As we saw webhooks earlier as an easy way to integrate with your product, enabling developers to create webhooks automatically on a user’s behalf helps reduce friction.” — Nicolas Grenié, Typeform

In terms of infrastructure, this means the payload must be structured in the same way as you define it, in the same order of questions.

To help enhance developer experience, TypeForm has since added a “Test Webhook” button on it platform. When someone puts in a URL, an option appears to test it first with a mock payload.

“No fake submissions — play with it until you’re ready.” Grenié says this “respects the same formatting and data types on your Typeforms.”

Now developers also have a way to look at all the events and all the payloads, what happened, and how the data was sent. He says this also carries over to debugging, to identify which side the fault is on.

To help developers, Typeform revamped their webhook panel last year to showcase both sides: the event sent by the Typeform platform and what your app sent back.

Grenié continued that webhooks-driven developer experience “lets you go back in time and see a history of all these events,” which allows for easy debugging.

VMWare Uses Webhooks to Crowdsource Code Samples

Richard Thomchick, a product manager at VMWare, spoke at DevRelCon 2019 in London about how his organization is leveraging webhooks to build a library of code samples. When he joined the company, they were distributing samples on CD-ROM. He said they “couldn’t keep up with the volume, variety, and velocity of code samples needed.”

That left people often swapping code samples in forums.

Thomchick led the development of a sample exchange as an aggregator, on a JAMstack solution — “like a Github users’ Github,” he said.

With this in mind, VMware set out to create contributor personas. This turned out to be not just developers and product teams but other upstream consumers that are less technical. Only about half of VMware’s developers are savvy with the git version control system.

“We knew we wanted all the content to end up on GitHub. Contributors that were Github savvy weren’t going to double work to add in the account. And the people that don’t have Github weren’t going to learn it to contribute a code sample,” he said.

Thomchick set up a sample exchange contribution workflow that abstracted the pull request process with a simple form to fill out and this pull request was automatically approved.

But then VMware suddenly lost its community manager who was broadcasting all of these new contributions. The development team used conditional statement chain app IFTTT so that if there’s a new item in the VMWare feed, it’ll automatically post a tweet about it. These automated tweets quickly became the ones with the most engagement.

Thomchick said that “When a sample is tweeted, it becomes a promotion platform and advocacy platform for the developer.”

VMware even created a new sample Exchange-as-a-Service for specialized channels. He says a REST API enables the creation of custom sample exchange for specific content. And you can test out those Webhooks samples within your VMware console. These are hand-selected by that product’s manager to highlight the highest quality and representative use cases.

Thomchick recommended other DevRel-focused teams create a Github organization, find the RSS feed and then connect to Twitter via a Webhook.

VMware is a sponsor of The New Stack.

Feature image by Myriam Zilles from Pixabay.