

بحث متعمق حول تقنية Webhooks

مقدمة

Webhook يُعدّ Webhook في تطوير الويب وسيلة لتعديل أو توسيع سلوك صفحات الويب أو التطبيقات عبر **استدعاءات رد (callbacks)** يحددها المستخدم . عندما تقع أحداث معيّنة في خدمة أو تطبيق، يقوم النظام بإرسال **طلب HTTP (عادةً POST)** إلى عنوان تحددته خدمة أخرى ليتم تنفيذ إجراء ما ¹ . قد يكون الحدث هو نشر كود في مستودع برمجي، أو إضافة تعليق في مدونة، أو وصول نموذج من مستخدم. ويمكن هذا الأسلوب المطوّرين من ربط خدمات مختلفة معًا بطريقة سلسلة وفي الزمن الحقيقي، وهو ما جعل **Webhooks** أحد اللبّات المهمة للهندسة المعتمدة على الأحداث في الويب.

النشأة والدوافع

من صاحب الفكرة؟

ظهر مصطلح *Webhook* لأول مرة عام 2007. المطوّر **جيف ليندسي** (Jeff Lindsay) هو من صاغ هذا المصطلح وروّجه ، حيث استوحاه من مصطلح *hook* المستخدم في البرمجة وأراد إيجاد آلية تسمح للتطبيقات بمشاركة البيانات دون الحاجة إلى الاستفهام المستمر (polling) ² . في تدوينته "Web hooks to revolutionize the web" وصف ليندسي الفكرة بأنها استدعاءات يحددها المستخدم ترسل عن طريق HTTP POST، حيث يسمح التطبيق للمستخدم بتحديد عنوان URL يريد تلقي البيانات عليه، ويقوم التطبيق بإرسال البيانات إلى هذا العنوان عندما يحدث الحدث ³ . أطلق ليندسي على هذا النوع من الإشعارات اسم "ويب هوك" وأشار إلى أنه يشبه إعادة توجيه *STDOUT* في واجهة الأوامر إلى برنامج آخر ³ .

لماذا ظهرت الويب هوك؟

قبل الويب هوك، كان الحصول على تحديثات من خدمة خارجية يتم غالباً باستخدام **تقنيات التغذية (feeds)** أو **استدعاء واجهات API بصورة دورية** . هذا الأسلوب يتطلب كتابة أنظمة جدولة لاستقصاء البيانات، والتعامل مع التخزين المؤقت والتحليل، وهو أمر مرهق ومكلف من حيث الموارد ⁴ . ليندسي رأى أن **الويب بحاجة إلى ما يعادل أنابيب Unix** —وهي آلية بسيطة لنقل البيانات بين أوامر مختلفة— لكن بصورة تناسب الويب الذي يعتمد على أحداث غير متزامنة ⁵ . لذلك اقترح **نظام دفع (push)** بدلاً من السحب، بحيث يتم إرسال البيانات فور حدوثها دون انتظار دورات الاستقصاء ⁶ .

كيف غيّرت الويب هوك القصة؟

أحدثت تقنية **Webhooks** نقلة في طريقة عمل الويب، إذ جعلت **التطبيقات أكثر استجابة** وسهّلت بناء تطبيقات متصلة. بدلاً من انتظار دورة الاستقصاء، تتلقى التطبيقات إشعارات فورية عند وقوع الأحداث. هذا مكّن خدمات مثل GitHub من توفير "تنبيهات ما بعد الالتزام (post-commit hooks)"، وقاد إلى ظهور خدمات تكامل مثل IFTTT و Zapier، وأصبح الوب أكثر اعتماداً على الأحداث. كما أشار ليندسي إلى أن الويب هوك يمكن أن تستخدم ليس فقط للإشعارات، بل أيضاً **لدمج تطبيقات مختلفة و تغيير سلوك التطبيقات عبر إضافات أو Plugins** ⁷ .

كيفية عمل الويب هوك

1. **تسجيل عنوان الاستدعاء:** التطبيق الراغب في تلقي الإشعارات يوفّر عنوان URL ليتم استدعاؤه. يُطلق على هذا العنوان الهدف أو *endpoint*.

2. **تحديد الأحداث:** يحدّد المستخدم أو المطوّر الحدث الذي يريده، مثل نشر مقال أو دفع طلبية. في Brightspot يُمكن للمستخدم تحديد نوع الحدث من قائمة الأحداث المتاحة ⁸.
 3. **الإرسال عند وقوع الحدث:** عندما يحدث الحدث في المصدر (مثلاً CMS ينشر مقالة)، يقوم المصدر بإرسال **طلب HTTP POST** إلى عنوان الويب هوك مع تحميل بيانات الحدث (payload). يُرسل الطلب عادةً بتنسيق JSON ³ ⁹.
 4. **معالجة الطلب:** يُعالج التطبيق المستقبل البيانات فوراً؛ يمكنه تخزينها، إرسال إشعارات، أو تشغيل عمليات أخرى. مثال: عند نشر تعليق في نظام إدارة المحتوى، يرسل الويب هوك إشعاراً لتطبيق Slack فيتم إخطار الفريق فوراً ¹⁰.
- هذا الأسلوب يعني أن **التطبيق المستقبل لا يحتاج إلى الاستفهام (polling)** للتحقق من وجود بيانات جديدة. بل يستقبل البيانات مباشرة عند حدوث الحدث، مما يقلل استهلاك الموارد ويزيد سرعة الاستجابة ⁴.

الفرق بين Webhook وواجهات API التقليدية

الجانب	واجهة API التقليدية	Webhook
آلية الحصول على البيانات	تعتمد غالباً على الاستقصاء (polling) ؛ يقوم العميل بطلب البيانات بصفة دورية (كل X دقائق مثلاً) ¹¹ .	تعتمد على الاستدعاء عند وقوع الحدث ؛ عندما يحدث الحدث، تُرسل البيانات فوراً إلى عنوان URL المحدد ¹² .
زمن الحصول على البيانات	قد يكون مؤجلاً حسب فترة الاستقصاء؛ يمكن جدولة فترات مختلفة ولكنها ليست فورية ¹³ .	فوري ؛ تُرسل البيانات في الزمن الحقيقي، ولا يمكن جدولة إرسالها لأنها مرتبطة بالحدث ¹⁴ .
استهلاك الموارد	يحتاج إلى خوادم وسيطة وجدولة، وقد يسبب حملاً على الشبكة للخدمات الكبرى.	أقل استهلاكاً لأن البيانات تُرسل فقط عند الحاجة، مما يقلل عدد الطلبات غير الضرورية ⁴ .
نماذج الاستخدام	مناسب للحالات التي لا تتطلب زمناً حقيقياً أو عندما لا تدعم الخدمة إرسال الويب هوك.	مثالي للتطبيقات المعتمدة على الأحداث أو التي تتطلب تزامناً فورياً مثل إشعارات الدفع أو تكامل أنظمة مختلفة ¹⁵ .

استخدامات شائعة للويب هوك

- **الإشعارات الزمنية** : إرسال إشعار فور نشر مقال أو تعليق أو طلب جديد. المقالة في Brightspot توضح مثالاً حيث يُرسل CMS بيانات المقال (العنوان، المؤلف، وقت النشر) إلى شاشة تليفزيونية بمجرد نشره ⁸.
- **تكامل الأنظمة** : يمكن استخدام الويب هوك لنقل البيانات بين نظامين؛ على سبيل المثال، إرسال بيانات نموذج إلى جدول في Google Sheets أو CRM، أو ربط خدمات مثل Slack و GitHub و Facebook ¹⁶.
- ليندسي اقترح استخدام الويب هوك لدمج تطبيقات مختلفة بإعادة تنسيق البيانات وتمريبها لتطبيق آخر ⁷.
- **تعديل التطبيقات** : يمكن استخدام الويب هوك لتمكين الإضافات (plugins) أو لإضافة خصائص جديدة لتطبيق قائم، بحيث تتيح للمطوّرين تعديل سلوك التطبيق حسب الحدث ⁷.
- **أتمتة المهام** : خدمات مثل IFTTT و Make تسمح بإنشاء سيناريوهات يتم تشغيلها عند وقوع حدث، مثل ملء جداول البيانات تلقائياً أو إرسال رسائل بريدية ¹⁷.
- **التكامل مع أدوات التطوير** : منصات مثل GitHub و GitLab تستخدم الويب هوك لتشغيل عمليات التكامل المستمر (CI) أو لتنبيه أنظمة تتبع العيوب عند حدوث التزام جديد في المستودع ¹⁸.

الجوانب الأمنية

نظرًا لأن الويب هوك يعتمد على استقبال طلبات HTTP من خوادم خارجية ، ينبغي اتخاذ تدابير أمنية لضمان صحة المصدر ومنع الهجمات:

- **المصادقة الأساسية (Basic Auth):** يمكن إرسال اسم مستخدم وكلمة مرور مع الطلب للتحقق من هوية المرسل ¹⁹.
- **السر المشترك أو التوقيعات الرقمية:** يمكن تضمين توقيع أو رمز سري في الطلب للتحقق من أنه قادم من المرسل الصحيح ²⁰.
- **توقيع HMAC:** العديد من المنصات (مثل Stripe و GitHub) تضيف رأسًا يحمل توقيع HMAC لمحتوى الطلب، مما يسمح للمتلقّي بالتحقق من سلامة البيانات ²¹.
- **المصادقة عبر TLS متبادلة (Mutual TLS):** يمكن استخدام شهادات TLS للتحقق المتبادل بين المرسل والمستقبل ²².
- **تثبيت عناوين IP:** يمكن للقوائم بالاستقبال السماح فقط للطلبات القادمة من عناوين IP موثوقة ²³.

عرض تقديمي في مؤتمر Conf42 حول "Secure and Scalable Webhooks" يشير إلى أن الويب هوك لم تُصمّم لتكون آمنة بشكل تلقائي، وأنه يجب توفير آليات تحقق مثل الرموز المشتركة وتوقيعات HMAC، واستخدام قوائم IP البيضاء لتقليل إمكانية استقبال طلبات خبيثة ²⁴.

أثر الويب هوك في بنية الويب الحديثة

منذ تقديمها عام 2007، أصبحت الويب هوك أداة أساسية للاتصال بين الخدمات. تقارير Google Trends (وفق عرض Conf42) تظهر زيادة كبيرة في الاهتمام بالويب هوك خلال السنوات الأخيرة ²⁵. تبنتها العديد من المنصات مثل GitHub و Slack و Stripe و Facebook لتوفير إشعارات فورية للمطورين. كما أصبحت جزءًا لا يتجزأ من الخدمات المؤتمتة مثل IFTTT و Zapier، والتي سمحت للمستخدمين غير التقنيين بإنشاء روابط بين التطبيقات بدون كتابة أكواد معقدة ²⁶.

تعزز الويب هوك الهندسة المعتمدة على الأحداث (Event-Driven Architecture)، حيث تُعدّ جزءًا من "البنية التحتية المفتوحة" التي تخيلها ليندساي؛ فهي تسمح بتوسيع التطبيقات بسهولة، وتجعلها قادرة على التفاعل في الزمن الحقيقي، وتفتح المجال لابتكارات جديدة في عالم الخدمات السحابية والأتمتة ²⁷.

خلاصة

تُعدّ Webhooks خطوة حاسمة نحو ويب أكثر تفاعلية. الفكرة، التي طرحها جيف ليندساي في عام 2007 لتجاوز مشكلة الاستقصاء وخلق ما يعادل أنابيب Unix في الويب، أصبحت اليوم جزءًا أساسيًا من معظم التطبيقات الحديثة ² ³. تعمل الويب هوك عبر إرسال طلبات HTTP POST عند وقوع أحداث محددة، مما يسمح بالتكامل بين التطبيقات وتوفير إشعارات في الزمن الحقيقي ²⁸. يُستخدم الويب هوك في إشعارات النشر وتكامل الأنظمة وتعديل التطبيقات وأتمتة المهام، مع ضرورة اتخاذ إجراءات أمنية كالمصادقة والتوقيعات الرقمية ²⁹. وبفضل انتشارها، ساهمت في دفع البنية المعتمدة على الأحداث والأنظمة السحابية، وجعلت من الممكن ربط التطبيقات والأجهزة والخدمات بسهولة وفعالية.

Webhook - HandWiki 29 23 22 21 20 19 18 9 2 1

<https://handwiki.org/wiki/Webhook>

Web hooks to revolutionize the web :: Jeff Lindsay 27 6 5 4 3

[/https://progrium.github.io/blog/2007/05/03/web-hooks-to-revolutionize-the-web](https://progrium.github.io/blog/2007/05/03/web-hooks-to-revolutionize-the-web)

What are webhooks, and how do they work? - Brightspot 15 10 8 7

<https://www.brightspot.com/cms-resources/technology-insights/what-are-webhooks-and-how-do-they-work>

What Are Webhooks and Why Would You Use Them? | Make 28 17 16 14 13 12 11

<https://www.make.com/en/blog/what-are-webhooks>

Conf42%20Golang%202023%20-%20Marvin%20Collins%20Hosea.pdf 25 24

<https://conf42.github.io/static/slides/Conf42%20Golang%202023%20-%20Marvin%20Collins%20Hosea.pdf>

What are Webhooks and How to Use Them? - IFTTT 26

<https://ifttt.com/explore/what-are-webhooks>