

Webhook

O webhook é uma técnica que vem ganhando seu espaço no mundo empresarial. Confira o nosso post e descubra: o que é, como funciona e por que sua equipe de tecnologia precisa experimentar. Já pensou se o usuário do seu software pudesse conseguir informações variadas em tempo real sem que ele precisasse interromper uma atividade, tendo de entrar em outras páginas e aplicativos para fazer buscas?

Seria demais, não?

Agora, imagine que ele seja capaz de poder conferir se o status das notas fiscais foram alterados diretamente do seu software, ajudaria bastante, não é mesmo?

Então... e seu eu te contar que esses são somente alguns diferenciais que a implementação do webhook pode proporcionar aos seus usuários?

Ficou interessado? Quer entender mais sobre o assunto? Então acompanhe o nosso guia para ficar por dentro a respeito desse recurso estratégico!

O que é Webhook?

O webhook, também conhecido como retorno de chamada web, é um método simples que facilita com que uma API forneça informações em tempo real sempre que um evento acontece.

Em outras palavras, trata-se de um modo de receber dados entre dois sistemas de forma passiva, ou seja, sem realizar nenhuma ação.

Portanto, um webhook é desenvolvido para reproduzir automações pré-definidas na ferramenta ou sistema de chegada, independentemente da origem da informação.

Dessa maneira, você poderá fazer com que uma ação X, do sistema 1, por exemplo, gere uma ação Y no sistema 2.

A ponta de “reação” das automações é fixa, mas cabe a você definir qual será o “gatilho” usado.

Assim, isso possibilita que programas simples e independentes sejam criados exclusivamente com o intuito de funcionar em cadeia conforme outros comandos são executados.

Qual a vantagem do webhook para os desenvolvedores?

Como vimos anteriormente, o webhook fornece automaticamente informações em tempo real, evitando que haja um trabalho de pesquisa a cada minuto em busca de uma resposta.

Por consequência, isso impede que aconteça uma sobrecarga do sistema e a perda de dados sempre que transcorrer algum erro.

Logo, um webhook facilita bem mais o serviço do desenvolvedor, além de deixar todo o processo mais simples e também mais barato.

Isso, pois, se formos analisar o que acontece em um sistema sem webhook, será possível identificar que fica impraticável, por exemplo, para uma integração descobrir onde enviar a mensagem.

Assim, a ferramenta fica obrigada a consultar periodicamente a API em busca de alguma mudança ou atualização.

Por fim, conclui-se que haveria um trabalho mais complexo e que exigiria muito mais tempo e empenho, além de encarecer o software.

Como funciona um webhook?

Webhooks permitem que sistemas externos recebam notificações de todos os eventos que ocorrem na API.

Quando um evento acontece, a API pode enviar uma requisição, sendo a mais comum a HTTP (POST), para a URL configurada no webhook com as informações relativas ao evento.

Ao receber a notificação, o sistema integrado à API pode executar diversas tarefas, dependendo do seu objetivo.

O webhook viabiliza que você receba notificações a cada alteração de status de um tipo de nota fiscal, por exemplo.

Desta forma, sempre que uma nota tiver seu status atualizado (aprovada, rejeitada, cancelada, etc) será enviado uma chamada via POST (PATCH, DELETE, PUT, etc), por exemplo, para uma URL pré-configurada conforme os parâmetros do seu conteúdo, de modo que você consiga realizar suas decisões comerciais.

Desse jeito, será possível criar gatilhos para determinadas ações, à medida que o status da nota fiscal for especificado.

Na prática, o que precisa ser entendido é que os Webhooks enviam dados em tempo real entre suas ferramentas ou sistemas para a URL do webhooks. E é dessa forma que você irá automatizar suas tarefas.

Diante disso, podem ser usados diversos formatos, sendo os mais utilizados, o JSON ou o XML. Veja só a diferença entre cada um deles:

- ➡ O **XML** é uma sintaxe básica que pode ser usada para compartilhar informações entre aplicações, permitindo reutilizar o código com objetivos diferentes no outro sistema ou ferramenta.
- ➡ O **JSON** é um formato mais leve de troca de informações entre sistemas. Suas siglas se referem a "JavaScript Object Notation", mas isso não significa que só pode ser usado com JavaScript.

Antes de criar um webhook para o seu sistema, é preciso que você observe alguns fatores de modo a evitar erros e problemas futuros.

Pensando nisso, montamos uma lista com os principais quesitos que você precisa se atentar antes de elaborar o seu webhook. Confira:

Tome cuidado com a perda de dados

Os Webhooks entregam dados ao seu aplicativo. Portanto, se a sua aplicação por algum motivo tiver um erro de dados, os mesmos podem ser perdidos.

Observe se há dados duplicados

Boa parte dos webhooks irão se basear nas respostas e solicitações de reenvio, sobretudo, se o seu aplicativo apresentar erros com frequência.

Se o seu pedido for processado e ainda houver esses erros, isso pode gerar dados duplicados em seu aplicativo.

Verifique se há erros de aplicação

Erros de aplicação são comuns, por isso, busque entender como o seu webhook provedor lida com as respostas de cada ação, assim você pode se preparar para contornar essa adversidade.

Explore a possibilidade da manipulação em escala

Um grande benefício dos webhooks, é que eles podem fazer vários pedidos ao mesmo tempo.

Portanto, o ideal é verificar se o seu aplicativo é capaz de manipular a escala esperada do seu webhook.

Opte por serviços adequados à sua demanda

A melhor maneira de compreender verdadeiramente um webhook é testar um serviço que se adeque à sua necessidade.

Felizmente, muitos serviços usam webhooks, assim, você pode facilmente aprender com eles e conhecer sobre todo o conteúdo referente ao sistema.

Webhook x API REST: quais são as diferenças?

Em uma API REST, definimos um recurso para receber dados, para então devolver alguma resposta.

Para enviar uma mensagem para um webhook, é feito uma requisição e também recebemos uma resposta (dependendo do caso).

Mas e aí?...Ambos fazem a mesma coisa? Qual é a diferença? Seria algo apenas conceitual ou de implementação?

Bom, vamos por partes: Um Webhook é um serviço, API é o conjunto de regras e métodos com que podemos comunicar e usar esse serviço.

Há webhooks que irão utilizar uma API do tipo REST, outros que usam API customizada, outras GraphQL, Thrift, etc.

Em outras palavras, um WebHook é um endpoint HTTP da qual um cliente deve implementar para receber notificações de eventos que ocorrem em um servidor.

Ambos fazem a mesma coisa?

Não. Em uma API REST, o servidor expõe um conjunto de serviços através de endpoints HTTP.

Já em um WebHook, o cliente expõe um endpoint HTTP e se inscreve no servidor para receber notificações de eventos.

Então qual é a diferença?

A diferença é que o cliente consome a API REST do servidor diretamente por meio de requisições HTTP, enquanto que os WebHooks são requisições HTTP das quais o cliente deve aguardar.

Como funciona o webhook do eNotas?

No caso do eNotas, seu sistema será notificado sempre que uma nota fiscal mudar para um dos status abaixo:

- Autorizada - Nota fiscal foi autorizada pelo órgão responsável;

- Negada - Nota fiscal foi negada pelo órgão responsável;
- Cancelada - O cancelamento da nota fiscal foi autorizado pelo órgão responsável;
- CancelamentoNegado - O cancelamento da nota fiscal foi rejeitado pelo órgão responsável;

Vale lembrar que, o valor em **negrito** representa o status que será retornado no parâmetro "nfeStatus".

Atenção: Parâmetro "nfeMotivoStatus"

Este parâmetro só retorna diferente de null nos seguintes status:

- "nfeStatus": "Negada";
- "nfeStatus": "CancelamentoNegado".

Já em relação aos dados do POST de notificação, estes serão variados conforme o tipo de nota fiscal.

No exemplo abaixo há o modelo de webhook para NFS-e:

Para cadastrar o webhook, será utilizado o painel do eNotas Gateway, acessando o menu Webhooks
→ Novo webhook:

Se seu sistema apresentar instabilidades ou estiver fora do ar, o **Gateway** tentará **lhe enviar o webhook de notificação até receber o status "HTTP 200"**, que indica que o webhook foi recebido e tratado com sucesso.

Serão realizadas **15 tentativas** com intervalos de tempo progressivos entre cada uma delas, **totalizando, em média, 4 horas**.

[illegible]