Webhooks assinados

Verifique a mensagem antes de processar o evento

Pré-requisitos

Conheça HTTP, eventos e o ciclo do Stripe Checkout. O exemplo usa Node.js e o SDK oficial da Stripe. Ele roda localmente com valores de teste e não chama endpoints da Stripe.

Preserve o corpo recebido [1]

A verificação usa o corpo original, o cabeçalho Stripe-Signature e o segredo de assinatura do endpoint. Se um middleware interpretar o JSON e depois serializá-lo novamente, os bytes podem mudar. Configure a rota para entregar o corpo bruto à função verificadora do SDK antes do processamento de negócio.

Use o segredo do endpoint correto

Cada endpoint e ambiente tem sua configuração. A chave da API autentica chamadas da aplicação à Stripe. O segredo do webhook verifica chamadas recebidas naquele endpoint. Mantenha-o na configuração protegida do serviço e siga o procedimento de rotação do provedor.

Considere o tempo e a repetição [2]

A biblioteca verifica a assinatura e aplica sua tolerância temporal. Mantenha o relógio sincronizado e preserve a verificação de recência. Uma nova entrega legítima pode ser assinada novamente. Verificar a assinatura continua necessário, e o processamento precisa lidar com repetições do mesmo evento.

Separe a verificação da decisão sobre a vaga

Depois de verificar a mensagem, confira se o tipo é tratado, se a sessão corresponde à tentativa e se o estado, valor e moeda atendem à regra. Encaminhe para processamento idempotente. Uma confirmação ao provedor deve refletir que o evento foi tratado ou registrado de modo durável para processamento.

Webhooks assinados

Exercício resolvido • Verifique o corpo original

Enunciado e dados de partida

Use um evento fictício com ID evt_local e tipo checkout.session.completed. Confirme que a biblioteca aceita o corpo original e recusa o mesmo conteúdo quando seu texto é alterado ou quando a configuração usa outro segredo. O exercício usa o gerador oficial de fixtures, sem uma sessão ou cobrança real.

1. Salve webhook.mjs

```
import assert from 'node:assert/strict';
import Stripe from 'stripe';

const stripe = new Stripe('sk_test_exemplo_local');
const segredo = 'whsec_exemplo_local';
const corpo = '{"id":"evt_local","type":"checkout.session.completed"}';
const assinatura = stripe.webhooks.generateTestHeaderString({
  payload: corpo,
  secret: segredo,
});

function verificar(payload, header, chave) {
  return stripe.webhooks.constructEvent(payload, header, chave);
}

assert.equal(verificar(corpo, assinatura, segredo).id, 'evt_local');
assert.throws(() => verificar(corpo + ' ', assinatura, segredo));
assert.throws(() => verificar(corpo, assinatura, 'whsec_outro_exemplo'));
console.log('Evento original aceito; alteracoes recusadas');
```

2. Prepare e execute no PowerShell

```
npm.cmd init -y
npm.cmd install stripe
node webhook.mjs
```

Webhooks assinados

Interprete recusas e retome com segurança

Resultado esperado

O terminal imprime ‘Evento original aceito; alteracoes recusadas’. Acrescentar um espaço depois do JSON mantém seu significado para um parser, mas muda o corpo que foi assinado. Usar outro segredo também impede a verificação.

Caso de erro • Middleware aplicado cedo demais

Se o teste puro passa mas a rota falha, revise o ponto que entregou o corpo ao SDK. Em homologação, confira se a rota recebeu bytes originais, se o cabeçalho está presente e se o segredo corresponde à origem da entrega. Evite registrar o segredo ou o corpo integral.

Variação • Evento conhecido, estado inesperado

Um evento válido pode chegar quando R45 já expirou. Nesse caso, a autenticação do evento passa e a regra da agenda impede a confirmação da vaga. Encaminhe para conciliação, conforme o estado do pagamento. Teste autenticação e política de transição separadamente.

Critérios para conferir

O resultado normal tem o ID esperado. Alterações do corpo e uso de outro segredo são recusados antes de qualquer efeito. O laboratório confere o SDK com uma fixture local. A validação da rota e do envio real deve ocorrer no ambiente de teste do provedor.

Referências primárias

[1] Stripe • Diagnóstico da verificação de assinatura

[2] Stripe • Recebimento e validação de webhooks

Webhooks assinados

Configure a verificação para cada ambiente

Preserve os bytes recebidos

A Stripe calcula a assinatura sobre o conteúdo recebido pelo endpoint. Configure a rota para entregar o corpo bruto ao SDK antes de fazer o parse. Cada destino tem seu segredo de assinatura. Em homologação, use o segredo do destino ou do encaminhamento de teste correspondente.

Exercício adicional

As entregas de teste chegam, mas todas falham na verificação. Por onde começar?

Resolução

Confirme o segredo do destino e se o middleware da rota preserva o corpo bruto. Envie um evento de teste pelo mecanismo oficial. Verifique que um evento válido chega ao processamento e que uma entrega com assinatura inválida é recusada.

Ao assistir ao vídeo

Observe a validação antes do processamento e o papel do corpo bruto.

Vídeo complementar

Stripe Developers - Webhook handling (em inglês)