Netlify + Stripe Webhook 签名验证失败

原因分析与完整解决方案

> 适用对象
>
> - 使用 **Netlify Functions**
> - 接入 **Stripe Webhooks**（一次性支付 / 订阅 / 发票）
> - 遇到签名验证失败、线上不通过、本地正常等问题

一、核心结论（必读）

Netlify 上 Stripe Webhook 签名验证失败，90% 原因不是 Stripe，而是：

> ❌ 传入 `stripe.webhooks.constructEvent()` 的 **不是原始请求体（raw body）**

Stripe 的签名算法对 **字节级一致性** 极其敏感，

任何一次 JSON 解析 / stringify / 编码变化，都会导致验证失败。

二、Stripe Webhook 签名机制简述

Stripe 会使用如下数据计算签名：

timestamp + "." + raw_request_body

只要发生以下任一情况，签名即失效：

- JSON.parse 再 stringify
- base64 未正确解码
- 请求体被中间件修改
- header 或 secret 不匹配

结论：

> `constructEvent()` 的第二个参数必须是「原始字节流」

三、Netlify Functions 的关键坑点

1️⃣ body 可能被 base64 编码（最常见）

在 Netlify Functions 中：

- `event.body` **有时不是原始字符串**
- 当 `event.isBase64Encoded === true` 时，必须先 decode
- 本地 `netlify dev` 和线上行为 **不一致**

👉 忽略该字段 = **线上必挂**

2️⃣ header 大小写不固定

Stripe 签名 header 可能是：

- `stripe-signature`
- `Stripe-Signature`

只取一种，会导致**偶发验证失败**

3️⃣ test / live webhook secret 混用

Stripe Webhook Endpoint Secret：

- Test：`whsec_test_...`
- Live：`whsec_live_...`

典型错误：

- 线上环境仍使用 test secret
- Stripe CLI 触发 test webhook，却校验 live secret

四、Netlify Functions 标准实现（推荐模板）

> ⚠ 不要自行简化此代码，这是验证稳定通过的版本

```
// netlify/functions/stripe-webhook.js
const Stripe = require("stripe");

const stripe = new Stripe(process.env.STRIPE_SECRET_KEY, {
  apiVersion: "2024-06-20",
});

exports.handler = async (event) => {
  if (event.httpMethod !== "POST") {
    return { statusCode: 405, body: "Method Not Allowed" };
  }

  // 兼容 header 大小写
  const signature =
    event.headers["stripe-signature"] ||
    event.headers["Stripe-Signature"];

```

```
  if (!signature) {
    return { statusCode: 400, body: "Missing stripe-signature header" };
  }

  // ✅ 关键：使用 raw body
  const rawBody = event.isBase64Encoded
    ? Buffer.from(event.body, "base64")
    : Buffer.from(event.body, "utf8");

  let stripeEvent;
  try {
    stripeEvent = stripe.webhooks.constructEvent(
      rawBody,
      signature,
      process.env.STRIPE_WEBHOOK_SECRET // whsec_...
    );
  } catch (err) {
    return {
      statusCode: 400,
      body: `Webhook signature verification failed: ${err.message}`,
    };
  }

  // 处理 Stripe 事件
  // checkout.session.completed
  // invoice.paid
  // customer.subscription.updated

  return {
    statusCode: 200,
    body: "ok",
  };
};

```

五、绝对不能做的事情❌

// ❌ 错误：JSON.parse 后再 stringify

const parsed = JSON.parse(event.body);

stripe.webhooks.constructEvent(

  JSON.stringify(parsed),

```
  signature,
  secret
);
// ❌ 错误：忽略 base64
const rawBody = event.body;
// ❌ 错误：只取一种 header
const signature = event.headers["stripe-signature"];
```

六、快速排查清单（照顺序执行）

✅ Step 1：确认 webhook secret

- 是否以 `whsec_` 开头
- 是否和当前 endpoint 对应
- test / live 是否一致

✅ Step 2：调试关键字段（仅调试用）

```
console.log("isBase64Encoded:", event.isBase64Encoded);
console.log("headers:", Object.keys(event.headers));
```

✅ Step 3：使用 Stripe CLI 对照测试

```
stripe listen --forward-to \
https://your-site.netlify.app/.netlify/functions/stripe-webhook

```

结果判断：

情况	结论
本地 OK，线上失败	base64 / env / header 问题
本地失败	raw body 已被破坏
CLI test 失败，Dashboard 成功	secret 混用

七、架构层面的经验建议

当出现以下情况，建议**迁移 webhook 后端**：

- 订阅逻辑复杂
- 多产品 / 多价格
- 多环境（staging / prod）
- 权限系统、幂等处理

推荐架构

- Vercel + Next.js API Routes
- 独立后端（Express / Fastify）
- Cloudflare Workers

> Netlify 继续用于前端部署即可

八、一句话总结

> **Stripe Webhook ≠ 普通 JSON API**
> **它是对“原始字节流”极度敏感的安全系统**

在 Netlify 中：

- 使用 raw body
- 正确处理 base64
- 使用正确的 webhook secret

👉 **签名验证一定能通过**

附录：常见错误速查表

报错信息	根因
No signatures found matching	body 被改写
Invalid signature	base64 / secret 错
Missing stripe-signature	header 取错
本地 OK，线上失败	Netlify base64