

[https://www.reddit.com/r/SaaS/comments/168regx/are\\_payments\\_the\\_most\\_difficult\\_part\\_of/](https://www.reddit.com/r/SaaS/comments/168regx/are_payments_the_most_difficult_part_of/)



**HappyCathode** · 1y ago ·

Yeah I get the feeling. Working on your code is kind of empowering, you own 100% of the process. And then, you need to integrate Stripe and you are bound to *their* processes. Like, you can't just make 1 API call to tell Stripe "bill this email 20\$ per month, thanks". You have to create a product, then a 20\$ price attached to it, create a customer, send them a checkout link, redirect, manage the customer coming back, maybe listen to some webhooks...

But at the same time, they abstract you a TON of financial and legal processes in +60 countries. However we may find Stripe complicated, it's way better than interacting with banks, credit cards and regulators.

And yeah, their documentation is all over the place. They support like 400 uses cases, which you have a need for maybe 1 or 2. It's kind of hard to filter out the noise.

I can highly recommend their official discord channel tho. Stripe support staff are super fast to answer and really helpful.

1 Reply Award Share ...



**Calebthe12B** · 1y ago

I agree, I don't get the hype around stripe documentation being so good. I found it confusing. It wasn't that the docs around endpoint were bad, they just didn't make it very clear which services needed to be used together. I have a subscription based service too, I finally got it to work using the following workflow:

- when a user signs up for my service, I immediately create a customer ID for them in stripe and store that with their user info. That's critical because every single step beyond this is going to rely on that id.
- you're going to need to set up one or more products in your stripe portal corresponding with your subscription plan. They're going to generate a 'price id'. Create a table in your database with those products and price ids. You'll need a way for your user to select the right price id on your webpage so you can send that over to stripe.
- from here, there's going to be one of two endpoints to send your user to. The first is a checkout session. A user who doesn't have a subscription already goes here. Stripe provides an endpoint you can use to check if that customer has any subscriptions with you. The checkout session will generate a URL you can redirect your user to, and then redirect them back to your site when they're done.
- once a customer ID is a paying customer, they need a way to manage billing. There's another user management/billing endpoint that I call whenever a user navigates to my billing page that generates a temporary link for your user to navigate to on stripe to manage their subscription.

On my phone, so don't have easy access to all of the exact endpoints you would need, but hopefully this can get you down the right track. I promise it's not you, it really is that confusing!

52 Reply Award Share ...

<https://x.com/levelsio/status/1853529236600176863>



@levelsio

Subscribe ...

Crazy bug on Photo AI today, and took hours to solve it:

> invoice.paid - No Photo AI user in db exists with Stripe customer ID: cus\_12345678

Stripe webhooks or payments (?) seem to have increased in speed so much now that it's started causing problems for me for about 1% of signups with this funny race condition:

1) customer.subscription.created arrives when a new customer signs up: they're subscribed to a plan, at that time I create a new user in the db for them, send them a welcome email with a login link, etc.

Sometimes though customer.subscription.created does NOT mean the payment actually went through, so you can't rely on that or you're giving users the product for free, so you have to wait for invoice.paid to arrive

2) invoice.paid arrives when the payment went through, at that point we know we got their money, so it loads the user account with photo credits and they can start using it

On a regular card payment this process happens in milliseconds, so you don't notice it, you just sign up and immediately you can start using the app

But 1% of time, the invoice.paid webhook arrives SO fast, that the customer.subscription.created webhook script is still busy CREATING the user, so it errors with:

> invoice.paid - No Photo AI user in db exists with Stripe customer ID: cus\_12345678

Because there is no user in the db yet, but a few milliseconds after there will be

My fix? Add sleep(1) to invoice.paid, it will wait 1 second and then it's definitely created

If even then not, it will reply with HTTP error 400, so Stripe can auto retry the webhook!

```
68 // <invoice.paid>
69 //
70 // this event is important because when user creates/changes subscription, but payment does not go through
71 // the subscription.updated or subscription.created event hits, even if unpaid
72 // only when invoice.paid event hits, we should actually add credits to their account
73 // then it's paid for
74 //
75 // {{ $event->type == 'invoice.paid' }}
76 //
77 // <sleep to avoid race condition>
78 // 2024-11-04: sleep 1 second to avoid customer.subscription.created creating new user in db
79 // being outran by invoice.paid here that adds the credits
80 // do wait 1 second
81 sleep(1);
82 // </sleep to avoid race condition>
83
84 $invoice = \Stripe\Invoice::retrieve($payload['data']['object']['id']);
85 $customer = \Stripe\Customer::retrieve($invoice->customer);
86
87 $query = $usersDb->prepare("SELECT * FROM users
88                                WHERE
89                                stripe_customer_id = :stripe_customer_id");
90
91 $query->bindValue(':stripe_customer_id', $customer->id);
92 $query->execute();
93 $user = $query->fetch(PDO::FETCH_ASSOC)[0];
94
95 // add credits after successful payment
96
```

12:06 PM · Nov 4, 2024 · 243.2K Views

<https://news.ycombinator.com/item?id=29584772>

**Hacker News** [new](#) | [past](#) | [comments](#) | [ask](#) | [show](#) | [jobs](#) | [submit](#) login

▲ Ask HN: Why is custom stripe integration so hard?  
1 point by brunosalic on Dec 16, 2021 | [hide](#) | [past](#) | [favorite](#) | 1 comment

I'm working on a custom stripe subscription integration for my SaaS web app for more than a month now. I implemented the same thing 5 years ago, and it somehow feels a lot harder now.

The biggest part of added complexity is "SCA" (like 2FA for payments) which was enforced to the whole industry by the EU regulation. Writing custom JavaScript is now required to get custom integration working :/

But, despite this external complexity, I'm also constantly struggling with Stripe's API and the overall approach. Their docs are terse and concise - written like manpages: you have to have an overall understanding of the system to use them effectively. I turned to their tutorials and videos, but they're very simplistic: they explain only the most basic scenarios.

I find myself doing a lot of manual testing - using stripe's API with a variety of scenarios just to figure out how things work. Then, once I understand a certain feature I have to figure how it works with the rest of the integration.

Example feature: "only add product to the subscription once it's paid for". Not even Stripe support developers knew how to handle this correctly - they did point me to the right direction, but I had to do a lot of manual API testing.

TL;DR I feel Stripe's API became too complex, and too "low-level" to work with directly. I often wish there was a library that would handle all the communication with Stripe. Or at least an extensive video-course that would get you very familiar with all the scenarios.

Am I the only one that has this impression?

▲ oblib on Dec 16, 2021 [-]

I looked at stripe years back when it was first trending and getting a lot of buzz and even then it looked to me like it was way too complex when compared to what I was doing with PayPal, which is simply setting up an account for new users or updating the timestamp for user renewals, and adding a PayPal button for their use on documents they create with the apps.

I've heard a lot of complaints about PayPal over the years but I've never had a problem with them. But I have seen a lot of complaints about Stripe the past couple years now, so you're certainly not alone trying to figure what they need from you.

<https://www.indiehackers.com/post/stripe-integration-was-hard-7faaf53bd3>

# Stripe integration was hard 🤔

by Onemorething

Last week I started (and finished) the Stripe integration for [findrailwork.com](https://findrailwork.com)

This isn't a subscription-based service, but instead one-time purchases that make a job posting "active". This doesn't seem to be that unique, but I still had a heck of a time finding relevant examples.

I tweeted at the Stripe Dev team to ask for help and they actually pointed me towards a few good resources and offered personal help! I was shocked at how responsive their dev advocates are.

But I still can't help but feel bad about how much of a challenge the integration was for me. How common is this? Did it seem easy for the rest of you?

—Onemorething posted to  Building in Public on May 30, 2021