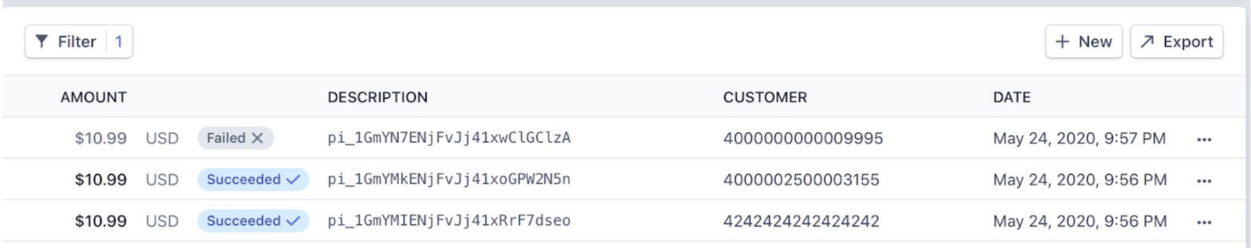
Evaluation Criteria -- Caroline Frost

1. Your program is able to easily run through the test cases on Stripe.com

This *appears* to be case (see picture below -- each customer is the cc number that was given in the guide, and the results were all the expected behavior), but unfortunately the guide did not update to reflect the tests had been completed. I assume the code for the metadata flag was perfect because I had copied it from earlier in the guide -- I couldn't find any documentation as to why it wasn't working.



AMOUNT			DESCRIPTION	CUSTOMER	DATE	
$10.99	USD	Failed ✕	pi_1GmYN7ENjFvJj41xwClGClzA	4000000000009995	May 24, 2020, 9:57 PM	•••
$10.99	USD	Succeeded ✓	pi_1GmYMkENjFvJj41xoGPW2N5n	4000002500003155	May 24, 2020, 9:56 PM	•••
$10.99	USD	Succeeded ✓	pi_1GmYMIENjFvJj41xRrF7dseo	4242424242424242	May 24, 2020, 9:56 PM	•••

2. The integration should keep a registry of all the successful payments that you would need to fulfill. You can append a line to a log file for every successful payment. You can use the Stripe CLI to test that webhooks work.

I used the 'fs' module for the logging bit, and logged to files named 'payment_success_logs' and 'payment_failure_logs' at the root. Sent some succeeds and failures to ensure everything was logging correctly -- you can see the output in my project folder.

3. README on your app (please provide instructions for the reviewer of how they should set up and test your implementation -- assume your reviewer has no experience in the language / framework that you choose)

Added in the project, but basics here as well:

- Used react js as requested, supplemented with cors and express for easy routing.
- Add your stripe secret key and public key in the index.js file; I removed mine.
- On executing 'npm start' from within root directory, client defaults to running on port 3000

- On executing ‘node src/server.js’ within root directory, server defaults to running on port 3001
- You can check that the client is running by going to ‘http://localhost:3000/’ in your browser; same with trying an endpoint on the server like ‘http://localhost:3001/create_payment_intent’
- To test the webhooks: execute ‘stripe listen --forward-to localhost:3001/webhook’ in the terminal and start the server. Execute ‘stripe trigger payment_intent.succeeded’ and ‘stripe trigger payment_intent.payment_failed’ in a new terminal tab, and you should see 200s and console logs.
- I used the ‘fs’ module for the logging bit, and logged to files named ‘payment_success_logs’ and ‘payment_failure_logs’ at the root. Sent some succeeds and failures to ensure everything was logging correctly -- you can see the output in my project folder. When you execute stripe trigger payment_intent.succeeded’ and ‘stripe trigger payment_intent.payment_failed’ there should be new lines added to those log files.

4. We are not evaluating the design of your website.

Probably for the best!

5. The friction-log is equally as important as the integration. We look forward to reading more of your feedback on the integration experience and any thoughts to the product design.

- **In summary, everything was extremely smooth sailing, except:**
  - I found it confusing that the documentation seemed to make some strange choices in the layout of the code, for example it seemed to split the /secret call and the /create-payment-intent calls. It was also strange that the some of the code in the documentation did not make it onto the Github project.
  - I’m pretty sure I did these correctly, but couldn’t get the green checks when logged in for the test cases on Stripe.com. Either I did it right and the testing was down, or the documentation wasn’t there for me to understand what I was doing wrong, but I’d like to have known for sure before turning this in. I was really bummed about it, I wanted to pass those tests as my reward for completing the integration!
  - The documentation for webhooks for testing was spread out across so many links; I really think these could be consolidated.
  - Lastly--it’s been a few years since I used React, so there were a few changes I had to get caught up on. Even so, I wouldn’t have been able to complete the assignment so quickly without having a background in React (took me several

        hours to do in total). Perhaps there could be other suggested options/github starter projects in the assignment Google doc?

    - **And I liked these parts of using the integration the best:**
        - The pre-made code, especially the css and forms. It made it very quick to get to a testable product, and it honestly did not look half bad considering I did not touch any css.
        - The dashboard was great overall: it had all the info I needed to look at for testing. That being said, I needed to click into each customer to view which credit card I had been testing at the time. For the given test credit card numbers, it'd be nice to flag those numbers at the top level. I ended up getting around this by setting the credit card number as the customer name such that I could see it at the top level.
        - Too much documentation and information is better than too little, and for the most part I was never lost. Just kept reading and following the guides.
    - **Full log:**

- Installed stripe successfully on computer, like the premade shell commands
- Signed up for an account
- Like the helpful security tips
- Like that toggling between languages on the documentation page causes all the premade shell commands to update to that language
- Seems strange that in the given code the create payment intent call isn't wrapped in its own async function
- I don't know how to use that given code without putting it into an async call, so I'm just going to create one. Not sure if that's the right way to go.
- Haven't used Express before but it seems like I should give it a shot. Seems like everyone uses it.
- Ah, is /secret supposed to contain the create-payment-intent call? That makes sense, and I think that's what the comment means. But then why didn't they just write it that way?
- Still not sure if I'm doing that correctly.
- Helpful to have the premade Javascript for the client side (since I've forgotten all my Javascript), but strange that we aren't updating some state for the client secret in this code block. Am I not supposed to be using some global state to keep track of things? Going to anyway after some Googling, since I'm not sure how else you would do this. My React knowledge feels very dusty when I learn ComponentDidMount is deprecated-ish.
- Adding Stripe.js, Elements and submitting Payment to Stripe was a breeze. Copied and pasted all that code, basically.
- Oh nice! They give us a test credit card number. I didn't want to use mine (?) and random numbers weren't working.
- Oh they give us more test numbers, and have some sort of test thing built into the documentation page that works if you're logged in, very cool.
- I did the tests and checked my code for the metadata flag and couldn't get it to work :(. No idea why, and very bummed about it. I'm assuming I did something wrong since I

couldn’t find any documentation or questions elsewhere on the internet about anyone else having problems.

- Ok let’s add some logging and try out the Stripe CLI.
- Odd that there are so many different places to read about webhooks/Stripe CLI -- I have four tabs open now that the documentation is linking me to and would rather just have one that contained all the step-by-step directions/was the source of truth.
- Embarrassing but true: don’t have homebrew installed since I recently updated my SSD and essentially wiped my computer.
- Brew complete.
- Opening more tabs -- being redirected from Github to https://stripe.com/docs/cli and then to https://stripe.com/docs/stripe-cli.
- Very nitpicky, but opened another tab -- the testing documentation is not what I wanted despite me clicking on a “webhooks” link, redirected to https://stripe.com/docs/webhooks. I think this is what I want.
- There is another tab that I opened as well that seems to be very similar. I really think the webhooks documentation could be consolidated.
- Oh no now I’m opening this https://stripe.com/docs/stripe-cli/webhooks. Do I already have that open? No I don’t. It seems like everything is up to date and agrees with each other, so that’s good at least.
- Nitpicky -- code given here https://stripe.com/docs/payments/payment-intents/verifying-status#webhooks doesn’t exactly match code here https://stripe.com/docs/stripe-cli/webhooks. Endpoint is /webhook in the first and /hooks in the second -- maybe host some of this example code on Github and embed from there.
- Super easy despite what the avalanche of documentation would suggest-- added the endpoint code, executed ‘stripe listen --forward-to localhost:3001/webhook’ in the terminal, started my server, getting 200s and logs to console. Webhook seems to be working for ‘stripe trigger payment_intent.succeeded’ and ‘stripe trigger payment_intent.payment_failed’. Didn’t need to add an endpoint to my dashboard even though some of the earlier documentation told me to do it.
- I used the ‘fs’ module for the logging bit, and logged to files named ‘payment_success_logs’ and ‘payment_failure_logs’ at the root. Sent some succeeds and failures to ensure everything was logging correctly -- you can see the output in my project folder.