

This is a short documentation on my process thought and approach to solving the problem.

1. Thought Process

I thought of building a database with tables for payment, buyer, seller, and the company and then apply the payment logic in my views.py.

2. Development

After reading Stripes documentation, I implemented the express checkout template to make building the system easier.

Installed stripe, created a checkout session with Stripe API. Built in my models tables for an event, buyer, company and seller and an admin page to handle these tables

The admin logs are **username:test password:Test**

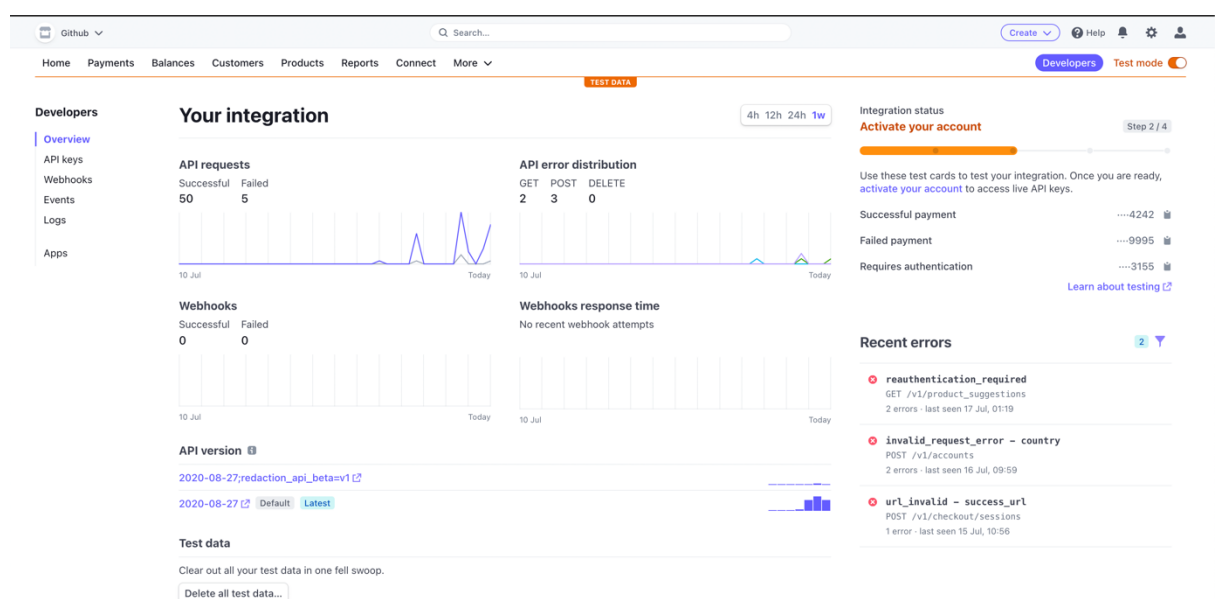
Some of the challenges I faced was

- Updating my database after Stripe payment occurs
- Fully implementing Stripe functionality to my code
- Forwarding payments to connected accounts

3. Test

With the help of Stripe API, I was able make a testcase where I created a random product. After payment and authentication is complete, funds are transferred to the company and then after commission balance is sent to buyer.

The diagrams below show the various test payments being made and the dashboard of the seller.



Payments

- All payments
- Collected fees
- All transactions
- Fraud & risk
- Invoices
- Subscriptions
- Quotes
- Payment links
- Orders

Payments

AllSucceededRefundedUncapturedFailed

ExportCreate payment

DateAmountStatusPayment method				Clear Filters	
	AMOUNT	DESCRIPTION	CUSTOMER	DATE	
	£100.00	Incomplete	pi_3LMaerGf400FY8Mo1RcGRPGg	16 Jul, 14:03	
	£100.00	Incomplete	pi_3LM6tJGf400FY8Mo8t1J2Rvx	16 Jul, 10:02	
	£100.00	Incomplete	pi_3LM6rEGf400FY8Mo895p11tn	16 Jul, 10:00	
	£100.00	Incomplete	pi_3LM6qqGf400FY8Mo8pp52nA	16 Jul, 09:59	
	£100.00	Cancelled	pi_3LL100Gf400FY8Mo85nPssZo	15 Jul, 11:05	
	£100.00	Succeeded	test@email.com	15 Jul, 10:37	
	£100.00	Cancelled	pi_3LLkv8Gf400FY8Mo80UfMdc	15 Jul, 10:34	
	£100.00	Cancelled	pi_3LLkv8Gf400FY8Mo8Uhlp6I8	15 Jul, 10:34	
	£1.00	Uncaptured	My First Test Charge (created for API docs)	15 Jul, 08:59	

9 results

PreviousNext