



What does Stripe do?

Because half of the fanboys have never even used the product

 Justin
Feb 1, 2021

11 2 ↗

The TL;DR

Stripe sells **payments infrastructure for internet businesses**: primarily, they help you bill your customers, process payments, and work with your payment data.

- Accepting payments on the web is a **pain in the ass** that requires **special bank accounts** and working with companies with names like SecureNet and Authorize.Net (yikes)
- Stripe gives developers a **set of simple APIs** for accepting and managing payments, and a **web interface** to explore and manage data manually
- Stripe has **expanded their product suite** over time to include broader financial services and data products

In the developer world (and tech more broadly), Stripe is known as the gold standard for how to market to and build products for a technical (read: software engineer) audience. They're also worth \$36B, and are expanding into broader financial services.

The core: payments

Stripe offers a lot of products, which can make it difficult to understand what they do exactly. To avoid that problem, let's pretend that we're a company called Dubstack that lets people create and write newsletters (just an idea off the top of my head). In order to let our users get paid for their content, we need to do a bunch of things:

- Be able to store and charge a credit card number
- Allow users to select monthly or yearly plans, and charge them every interval
- Generate and send invoices (if so requested)
- Work with payment data ad hoc (refund a customer, etc.)

This is a lot of work. And it's not the kind of developer work that's *straightforward* and time consuming - it's *difficult* and time consuming. Your typical developer doesn't have experience working with payment processing, and your typical early stage startup doesn't have a finance guru who can help that developer.

Stripe takes care of all of this for you via really well designed APIs. You don't need to worry about storing data, charging cards, or managing subscriptions. Let's walk through a couple of examples of API endpoints that we might use in our day-to-day at Dubstack.

🧠 Jog your memory 🧠

An API takes input data and performs some sort of task for you, typically returning some new data. In Stripe's case, you might tell one of their APIs to create a new customer in their system, and give them an email address to use. For more info, read [the full What's an API explainer here](#).

🧠 Jog your memory 🧠

→ Charging a credit card

When a user signs up for a paid plan, we need to charge their card so we can pay the rent. Stripe's charge creation API lets you send a basic HTTP request with an amount and an optional customer ID.

```
curl https://api.stripe.com/v1/charges \
-u sk_test_4eC39HqLyjWDarjtT1zdp7dc: \
-d amount=2000 \
-d currency=usd \
```

All you need to do is make the request - Stripe takes care of everything else.

(Note: Stripe is migrating over to a new API called [Payment Intents](#))

→ Updating customer information

Instead of maintaining our own database of paying users and their card information, Stripe takes care of storing all of that securely, and gives us nice APIs to read and update that information. If one of our customers needs to change their credit card information (Citi Double Cash is giving 2% cash back on everything now so this is completely reasonable), we'd use Stripe's [customer endpoint](#):

```
curl https://api.stripe.com/v1/customers/cus_IH1SyIQldcZgPg \
-u sk_test_4eC39HqLyjWDarjtT1zdp7dc: \
-d "metadata[order_id]"=6735
```

Each customer in Stripe has a unique ID (cus_IH1SyIQldcZgPg in this example). To update that customer, we just need to find that ID, append it to the base endpoint, and send over what data we want to change. This API might power a simple form we have on our site that lets users input their new info.

→ Cancelling a subscription

If our loyal readers are dumb and want to unsubscribe from a paid plan, we need to stop charging them and refund their prorated balance. Stripe's [subscriptions endpoint](#) lets you do that too:

```
curl https://api.stripe.com/v1/subscriptions/sub_AfFjc0iXtLSAxo \
-u sk_test_4eC39HqLyjWDarjtT1zdp7dc: \
-X DELETE
```

Like in the previous example, we need to specify to Stripe which subscription ID we want to cancel. We can also get into the details and decide to invoice the customer for their remaining balance, or prorate the cancellation.

So basically Stripe takes care of all of this annoying stuff that we would have to build in house otherwise. The three examples we gave here involve taking *action* (i.e. they're POST requests), but Stripe APIs also allow you to *read* (i.e. GET) your data and use it across your application.

⚠ Confusion Alert ⚠

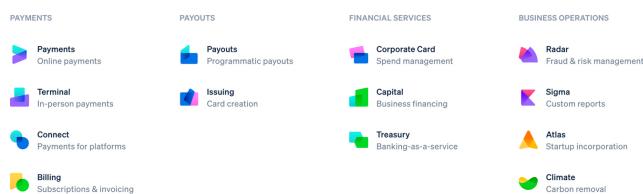
To use Stripe, you (generally - see below) still need to build a frontend on your site to actually ask your users for their name, email, credit card info, etc. But instead of storing that data yourself, you just send it right over to Stripe's APIs from that form. This is usually what it means to be an "API company."

⚠ Confusion Alert ⚠

Expanding product lines

Payments (accepting and managing card and customer info) and billing (subscriptions and invoices) are Stripe's two primary product lines, and if I had to guess, account for the overwhelming majority of their revenue. But as any company does as they grow (a lot), Stripe has released new product lines that integrate with each other and draw companies like Dubstack further into their ecosystem.

On their site, Stripe organizes their products into 4 categories:



I think it's actually simpler to organize them into 3 categories:

1. Payments

In addition to payments and billing, Stripe also offers:

- Terminal - a physical POS system to use in your donut shop or whatever
- Connect - lets you send payments out to your own customers (imagine you're running a marketplace)
- Payouts - lets you programmatically pay out your freelancers, users, or whatever have you (some overlap with Connect here that I don't understand)

Over the past 1-2 years, Stripe has released two new lil' products under payments that I think are very important:

- Checkout - a fully out of the box, Stripe hosted checkout page for taking in credit card info from new customers
- Portal - a fully out of the box, Stripe hosted billing portal where customers can update their info and adjust / cancel their subscriptions

Generally, you need to build the frontend (a web page with a form) that you *back* with the Stripe APIs - these two products mean you barely have to build *anything*. These haven't gotten a ton of fanfare yet, but I've used them personally and they're pretty awesome if you're resource constrained.

2. Financial Services

This has been Stripe's most experimental area.

- Issuing - an API for creating credit cards (physical or digital)
- Corporate Card - lets you create and manage corporate cards for your company
- Capital - access to debt financing for growing businesses
- Atlas - easy company incorporation and formation (LLCs, C-Corps, etc.)
- Treasury - banking as a service (hold funds, payouts, etc.)

The overlap between Issuing and Corporate Cards can be a bit confusing, but you can think of Issuing as a service if you want to create *external* cards (e.g. Venmo releasing a Venmo card), and Corporate Cards as a service for creating *internal company* cards (so you can expense the "friday" happy "hour").

As is often the case, each one of these product lines could be its own billion dollar business, which should be a reminder to all you kids of the power of product ecosystems when done right.

3. Working With Data

In addition to helping you do stuff - like charging cards and updating subscriptions - Stripe is also your system of record for payments. They store all of the data around who your customers are, what their payment info is, when they've been charged, and lots more like that. So naturally, Stripe offers tools for working with and understanding that data:

- Sigma - lets you write SQL against your Stripe data, and save and share queries with your team
- Radar - automatic fraud detection and alerting so you don't get scammed

These two products have been around for a while. Coincidentally, the marketing page for Radar has one of my favorite pieces of headline copy - "Use ML that actually works":



Use ML that actually works

Radar helps detect and block fraud for any type of business using machine learning that trains on data across millions of global companies. It's built into Stripe and requires no additional setup to get started.

Shots fired, Stripe copywriting team.

These categories don't map to what makes money for Stripe per se (you could organize products into loss leaders and money makers). They also recently added

Climate, which doesn't fit cleanly into any of these categories.

The UI, the pricing, and the fanfare

→ The UI / web interface

We've focused mostly on Stripe's APIs, and to be fair, they're what has gotten them this far - especially as a company focused primarily on serving a technical audience of software engineers. But Stripe also has a very nice web interface for when you need to search your customer base, quickly update a subscription, or check a payment for fraud in Radar.

I can take a look at Dubstack's active subscriptions:

The screenshot shows the Stripe dashboard for 'Subscriptions' under the 'Technically' account. The left sidebar includes links for Home, Payments, Balances, Customers, Subscriptions (which is selected), Invoices, Connected accounts, Products, Reports, Developers, and Settings. The main area is titled 'Subscriptions' with tabs for 'Current', 'Scheduled', and 'Canceled'. A search bar and filter options ('Filter', '+ New', 'Export') are at the top right. Below is a table of active subscriptions:

Customer	Status	Billing	Product	Created
paidtest@paidtest.com	Active	Auto	Technically - Monthly Subscription	Aug 29, 9:47 PM —
sessiontest2@gmail.com	Active	Auto	Technically - Monthly Subscription	Aug 23, 3:25 PM —
sessiontest@gmail.com	Active	Auto	Technically - Monthly Subscription	Aug 23, 3:04 PM —
sessiontest@gmail.com	Active	Auto	Technically - Monthly Subscription	Aug 23, 3:02 PM —
paid@paid.com	Active	Auto	Technically - Monthly Subscription	Aug 16, 10:31 AM —
paid@paid.com	Active	Auto	Technically - Monthly Subscription	Aug 16, 9:31 AM —
justin@justin.justin	Active	Auto	Technically - Monthly Subscription	Aug 15, 4:21 PM —
neer1@neer1.com	Active	Auto	Technically - Monthly Subscription	Aug 15, 9:47 AM —
yoon@yoon.yoon	Active	Auto	Technically - Monthly Subscription	Aug 14, 7:44 AM —
each@each.com	Active	Auto	Technically - Monthly Subscription	Aug 14, 7:04 AM —

And then cancel one if I don't like them:

A screenshot of the Stripe UI showing a list of subscriptions. The sixth row, belonging to 'justin@justin.justin', has a context menu open with the following options: 'Update subscription', 'Cancel subscription', 'Pause payment collection', 'View customer →', and 'View subscription →'.

Stripe also gives you a UI for viewing your customers, individual charges, invoices, and many other useful things like universal search.

→ Pricing

As with any developer focused startup with multiple product lines, Stripe pricing is complicated and confusing.

For payments, Stripe charges 2.9% + 30 cents for each successful charge. You'll also need to deal with separate pricing models for domestic bank transfers (0.8% with a \$5 cap), additional payment methods, and instant payouts. If you want Radar, that's 5 cents per screened transaction, *but it gets waived for accounts with standard pricing*. Sigma starts at 2 cents per charge (even though Sigma isn't used for charges, which is weird), and "infrastructure fees may apply," whatever that means. Oh, and you can pay for premium support, which *starts* at \$1,800 a month but you'll need to get in touch with sales to get an actual quote.

The screenshot shows the Stripe 'PAYMENTS' page. It highlights a 'complete payments platform engineered for growth'. It lists supported payment methods: Cards and digital wallets (VISA, Mastercard, American Express, Discover, Apple Pay, Google Pay), Domestic bank transfers, and ACH Direct Debit. It also mentions a \$5.00 cap for ACH transfers. The page provides detailed information about fees: 2.9% + 30¢ per successful card charge, +1% for international cards, and +1% if currency conversion is required. There are also sections for 'Additional payment methods' and 'See more payment methods'.

The screenshot shows a section of Stripe's pricing page. At the top, it says "Starting at 80¢ for iDEAL" with a link to "See more payment methods". Below this, there are several sections: "3D Secure authentication" (with icons for American Express, Diners Club, MasterCard, Visa, SEPA, and SOFORT), "Card account updater" (described as automatically updating expired or renewed card information for saved customers), "Adaptive Acceptance" (described as machine learning models that help increase revenue by improving authorization rates in real-time), "Instant payouts" (described as letting you access funds within minutes using an eligible debit card from the Dashboard), and a note about "Custom pricing available for companies with large payments volume or unique business models" with a link to "Contact Sales".

(This is the pricing sheet for just payments)

Stripe is expensive, but that value developers get from it is so colossal that they're not losing too many customers to "I went with the cheaper option." And pricing multiple products and services is very, very hard (I worked on this at DigitalOcean, it's not fun). So I don't necessarily blame them for this nightmare of a pricing page. Still, I wonder if a pricing calculator and some intelligently designed bundles could make all of this a lot simpler.

→ The Fanfare

If you work in tech, you probably already know that Stripe is *basically worshipped* in the community. Their APIs, site design, product detail, support, and documentation are generally held in quite high esteem. It's also considered a great place to work, to the point where "we're hiring at Stripe" has become a meme across Twitter. Daily, you'll see threads lauding the company from people who haven't even used the product – which to me seems...weird.

Now I'm no culture analyst - I'm just here to explain how the tech works. The important part is that Stripe's reputation among developers - their core audience - is rock solid, and as a user myself, their APIs and documentation are indeed quite excellent. So generally, if you're trying to cut through the noise and understand if something is actually good or not, look at what the actual users are saying.

Further Reading

- A guide on how to accept internet payments if you're *not* using Stripe is a good advertisement for Stripe
- There are plenty of Stripe alternatives that generally fly under the radar but should get more attention, like Adyen
- Stripe's guide to PCI compliance does a good job explaining why building your own systems is so hard

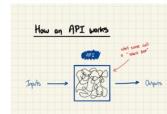
11 2 Share

Write a comment...

Ben Feb 2, 2021 Liked by Justin Gage
Hey Justin, another great article. Can I request Coinbase sometime? Thank you!

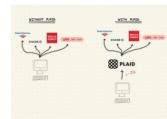
3 Reply

Nat Dec 7, 2021
Really like your articles. Can you go deeper about the legacy problem they solved? How they connect to the banks? (Authorize.net etc)
 Reply

**What's an API?**

What McDonalds and Lyft have in common

Justin Jan 9, 2020 ❤ 187 ⚡ ↗

**What does Plaid do?**

Technically begrudgingly tackles Fintech

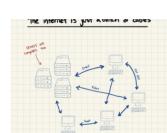
Justin Jan 14, 2021 ❤ 34 ⚡ 3 ↗

**What does New Relic do?**

Keeping an eye on your servers and apps

Justin Jan 11 ❤ 23 ⚡ 2 ↗

Company id: org.50...
People: 1
Monthly spend: \$0.00
Company website: Unknown

**What's Reverse ETL?**

Getting your data OUT of your warehouse?

Justin Feb 1 ❤ 19 ⚡ ↗

**What happened to Facebook?**

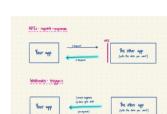
A basic explainer of what outage was all about

Justin Oct 5, 2021 ❤ 29 ⚡ 4 ↗

**What does GitLab do?**

The TL;DR GitLab is a somewhat contrarian take on DevOps: it's basically one giant tool for literally anything you'd want to do relating to building and...

🔒 Justin Jan 4 ❤ 8 ⚡ ↗

**What's DevOps?**

IT has a cool new name

Justin Jan 5, 2021 ❤ 34 ⚡ ↗

**What are webhooks?**

Triggered

Justin Sep 13, 2021 ❤ 28 ⚡ 5 ↗

**What's Headless E-Commerce?**

We may be running out of names

Justin Nov 2, 2021 ❤ 20 ⚡ 7 ↗

**I built a (basic) Substack clone in a month**

This was probably a waste of my time

Justin Sep 2, 2020 ❤ 50 ⚡ 4 ↗

[See all >](#)