/dev/finance inc.

finance for developers, by developers

Mission

It's slow, complicated and expensive to start accepting payments online. Talented software developers can launch products quickly—the first version of Facebook was written in a week—but setting up the infrastructure to collect money from customers can take months.

> "Time to build a useful web app: three weeks. Time to add subscription billing: three months. If someone could make that easier? Priceless."
>
> — Andrew Catton, DabbleDB

> "We want to implement subscription billing. We planned to do it six months ago, but gave up because of the complexity. We did an iPhone app instead."
>
> — Ross Boucher, 280 North Inc.

> "Building out our credit card processing system was harder than anticipated at every turn. I wish we could have instead applied those countless hours of development time to our core product."
>
> — Aston Motes, Dropbox Inc.

This opportunity for improvement hasn't gone unnoticed. Google, Amazon, PayPal, and Square all boast developments in this space, and it's widely known that Facebook are working on a new payments system. A recent development has shown what the future of payments will look like: when merchants sell their creations in Apple's iPhone App Store, consumers are charged by Apple. Apple interacts with banks, gateways and credit card processors on behalf of the merchant. The merchant's only interaction with the finance industry is the monthly payment they receive. Even with the advent of these changes, It's clear that no single solution has come close to displacing traditional payment processors.

/dev/payments—/dev/finance inc's flagship product—seeks to become the simplest way to accept payments online by managing all interaction on behalf of the merchant. The Internet is replete with "winner-takes-all" markets. eBay is the only auctioneer's tool that matters; Google, the only search engine; Facebook, the only social network. No entity has managed to become the default tool for accepting payments. /dev/finance inc exists to fix this.

Team

Patrick Collison is an Irish scientist, engineer, entrepreneur and alumnus of the Massachussetts Institute of Technology. At the age of sixteen he was the winner of the 41st Young Scientist and Technology Exhibition with the creation of Croma— a new LISP-based programming language. By age nineteen Patrick had negotiated the sale of his first company, Auctomatic (co-founded with brother, John Collison) for $5 million — a venture spanning 18 months from incorporation to sale. Patrick has given countless talks on subjects far ranging from programming language design to private equity and is readily considered an expert in his field.

John Collison is an Irish scientist, engineer and entrepreneur. At age sixteen, John co-founded Auctomatic, an eBay competitor, with his brother Patrick. Starting from humble beginnings in the National Technology Park in Limerick, the two left Ireland for California where they built the company out of Silicon Valley—leading to a successful acquisition months later. John has won multiple gold medals at Irish science olympiads and completed his schooling with 980 points in Ireland's university entrance exams—giving him the highest score in all of Ireland to date by a margin of 10%. John studied physics at Harvard University.

Raeez Lorgat is a South African scientist, engineer and entrepreneur. He was the computer science grand award winner of the 2007 Intel International Science and Engineering Fair, and winner of the Eskom Expo for young scientists at the age of fifteen. Alumnus of many national and international science and informatics olympiads, he has given talks ranging from the design and analysis of algorithms, to the impact of government policy in educating the developing world. At age sixteen, NASA and MIT's lincoln labs honored Raeez with the naming of Minor planet 23122 Lorgat. Raeez studied mathematics and computer science at MIT.

/dev/finance inc.

finance for developers, by developers

The Merchant Experience

A large part of what makes /dev/payments offering unique lies in the philosophy that shapes its services. Where traditional payment processors expect each merchant to develop both an intimate understanding of the industry and the ability to navigate its individual components, /dev/payments takes the opposite approach; a stance driven by an understanding for the need for simplicity and the growth it enables. A /dev/payments-empowered merchant needs only to interact with a /dev/payments account—requests made to /dev/payments facilitates the total transfer of money from customer to merchant. By providing a controlled, low-barrier payments service, we can deliver a first-class experience to our trusted merchants, stripping the obstacles standing in the way of enabling growth; much in the same way mobile app-stores have empowered developers in ways previously unimaginable.
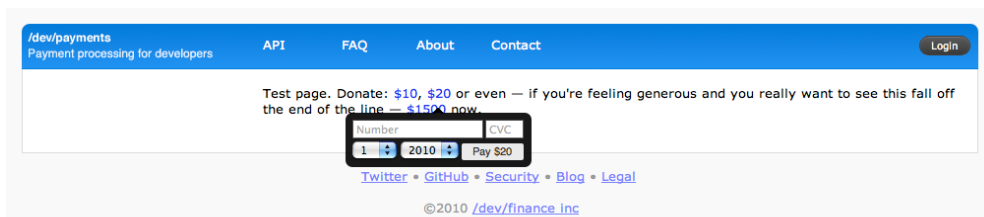
CREDIT CARD PAYMENTS THROUGH A POPUP WIDGET

The Developer Experience

/dev/payments provides its services through an evolving set of interfaces; the core of which lies in the /dev/payments API — a modern, simplified set of payment interfaces enabling the generic processing of payments. By taking into account the common needs of today's web-applications *and* their developers, /dev/payments is able to anticipate and prevent the duplication of commonly-implemented routines — a bane of any developer integrating payment systems online today. Parallel to the API, /dev/payments provides the necessary tools granting developers the option to fully customise the payments experience presented to the consumer. As such, a /dev/payments developer can meld /dev/payments to their needs; customising the full spectrum of functionality and form, from details such as aesthetics to specifying the policy behind handling sensitive data. Peripheral to this, /dev/payments provides a unified merchant interface— along with the tools for each merchant to build their own— allowing for account management, technical settings and financial performance analytics.

```
# this actually works - paste into a shell
$ curl https://api.devpayments.com/api \
    -d method=execute_charge \
    -d 'card[number]=4242424242424242' \
    -d 'card[exp_month]=10' \
    -d 'card[exp_year]=2011' \
    -d amount=300 \
    -d currency=usd \
    -d identifier=bob@gmail.com \
    -d key=G5m5CTNrfS5uwtpUCniFPuwbiODkHn
```

EXAMPLE /dev/payments API CALL

Payment processors today   fail to do justice to the fast-changing landscape surrounding online payments; a fact immediately evident to any software developer targeting the modern consumer-oriented web. The status quo is such that it is easier for a developer to integrate payments into a mobile application then it is for any modern website to do the same—a situation that has led to the birth of the current trend of 'monetization through a mobile application' as seen today in many web-applications. That it is still easier to accept payments in a mobile application than through a web-browser is an anachronism—and /dev/payments changes this.

/dev/finance inc.

finance for developers, by developers
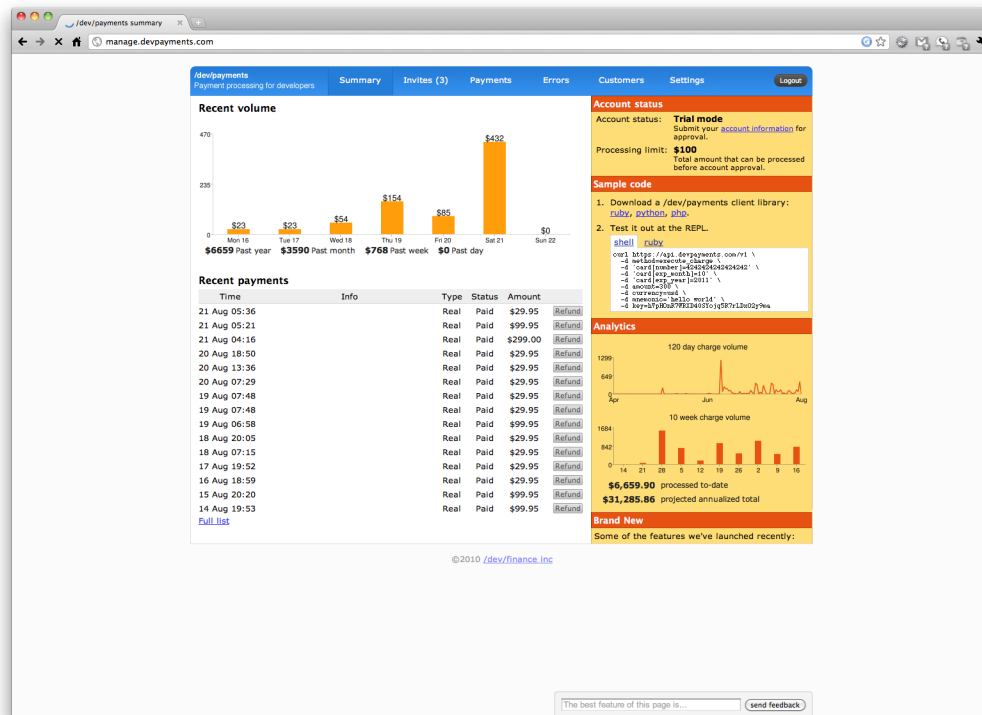
The problem of Distrusting Merchants

Conventional wisdom describes the attitude towards which all online merchants are treated; an approach emblemized by the account creation process found today with all traditional payment processors—often a lengthy and testing process. Merchant distrust has been a necessary consumer protection in a marketplace strewn with both legitimate and questionable merchants; and it's often very difficult to discern which. This problem of 'distrusting merchants' leads to a poor experience—at least initially—for merchants looking to charge for online products and services.

Trust and the Merchant Network

Where merchants today are treated as 'fraudulent until proven trustworthy', /dev/payments forgoes this traditional methodology in favour of implementing a trust network between all /dev/payments merchants. By constraining the availability and distribution of /dev/payments payment processing accounts to existing pre-approved /dev/payments merchants, we can ensure that only trusted merchants have the ability to bring other trusted merchants online. This method of 'invitation through referral' ties each new merchant to an existing /dev/payments merchant — the positive incentives to invite new merchants are balanced by negative incentives of associating one's company with potentially untrusted merchants. The result; trusted merchants inviting other trusted merchants; a social network with growth balanced on the personal risk of associating the inviter with potentially non-trustworthy invitees.

Much of the difficulty found with fighting fraud under the current system is as a direct result of the inability to accurately identify merchants; fraudulent merchants both masquerade under false identities and boast fraudulent relationships. By building a detailed social graph of merchants, /dev/payments is better equipped—from the start—to fight fraud.

/dev/finance inc.

finance for developers, by developers

Merchants using /dev/payments

280 North inc.
http://280north.com

Posterous inc.
http://posterous.com

Anybots Inc.
http://anybots.com

Simperium inc.
http://www.simplenoteapp.com

inDinero inc.
http://indinero.com

Treefly Inc.
http://www.snipshot.com

MockingBird inc.
http://gomockingbird.com

Seeing Interactive inc.
http://seeinginteractive.com

NewsLabs inc.
http://newslabs.com

General Projects
http://generalprojects.com/

Ginzamarkets, Inc.
http://ginzametrics.com

Clickpass inc.
http://clickpass.com

Providing the Solution

/dev/finance inc. provides the right answer to an imminent gap in the payment provider landscape. It does so by integrating the solutions to two distinct problems. Take the complex mesh of gateway agreements, merchant account providers and resellers coupled with tiered fee structures and an industry coupled with mountains of regulation, and it is obvious to see why very few, if any, developers have the ability, resources or willpower to tackle the problem at large. On the other hand, merchant banks, payment processors and gateways that do have the ability to provide the financial infrastructure lack the context to deliver a product and service tailored to serve the modern developer. /dev/finance inc. presents a fusion of finance and technology; this comes at a time where none can doubt the growth potential in the enabling of talented developers. The goal is simple: to become the default payment processing solution for any and all software online. For /dev/payments, this translates to providing nothing but a first-class experience to merchants and developers.