Developer Onboarding: GitHub

This is the fifth in a series of articles focussing on reviewing Developer API Onboarding. In this post I'm looking at the API onboarding offered by GitHub.

Criteria

I will be looking at 3 aspects of the onboarding experience:

1. **Exploration**: Does the experience help me to understand or try out their product before signing up for an account?
2. **Onboarding & Integration**: How well does the product guide me from their main site to create an account and to make that first API call or integration?
3. **Reference**: Once I have my first integration in place, how hard is it to find the full reference documentation for the API call I made?

Additionally I will look at the **emails** sent by GitHub after I sign up.

GitHub

A better way to work together

betta.io

Developer Experience: GitHub READMEs

📅 February 7 2017   developer experience   developer onboarding   dx highlights

The `README.md` for a project on GitHub is often the first thing developers see when they encounter your product. The README for a library can be the starting point in a developers relation with your product. These text files serve both as a technical guide to your library and as marketing material for your product. So what should be in a README and what is better avoided?

An analysis

For this article I've done a little survey across the SendGrid, Pusher, Stripe and Twilio Ruby libraries. I've also taken a screenshot of each README and put them in a handy little gallery for future reference.

	SendGrid	Pusher	Stripe	Twilio
Repo name	`sendgrid-ruby`	`pusher-http-ruby`	`stripe-ruby`	`twilio-ruby`
Build status badge	✅	✅	✅	✅
Library version badge	❌	❌	❌	✅
Introduction	129 words	0 words	100 words	19 words
Get Started samples	✅	✅	✅	✅

betta.io