Dylan Slack https://dylanslack.com https://github.com/dylnslck dylanslack@gmail.com (303) 325-6597

Design-oriented full-stack developer looking for a senior engineer position at a fast-paced, quicky scaling startup where there are opportunities to grow and learn, implement the latest (yet pragmatic) tech, evangelize the company, and build awesome, customer-loving products.

Recent experience / projects

Co-founder and CTO of Directly.io

August 2014 - Present

Mocked, designed, and implemented Directly's consumer-facing and business-facing products. Led an agile team in the construction of a custom RethinkDB-ORM powered backend that served complex business data to Directly's React + Redux frontend.

Lead developer at Coerver Colorado

May 2013 - Present

Designed Coerver's Node.js + GraphQL powered registration and payment backends that have processed millions of dollars worth of transactions. Led the development of Coerver's MongoDB powered analytics system, which has stored and analyzed hundreds of thousands of data points.

Top 40 at Jason Calacanis' Launch Hackathon

February 2016

Travelled to San Francisco and designed a CLI-based API generator and auto-scaler with an auxillary Ember.js GUI. Built on top of AWS and Heroku.

Co-founder of Snupol

May 2012 - January 2014

Learned the LAMP stack in May 2012 and launched a collegiate textbook marketplace three months later. Snupol scaled to seven universities, made some hires, but ultimately dissolved for various reasons. Snupol served as my first entrepreneurial failure.

Academic tutor at Georgia Tech

August 2013 - December 2014

Tutored undergraduate calculus, computer science, physics, and electrical engineering coursework at Georgia Tech's Center of Academic Success.

Education

Georgia Institute of Technology

B.S. in Electrical Engineering
Summa Cum Laude honors (3.86 GPA)

Skills / tools / platforms

specialize | pretty good | currently learning

Backend

Node.js / ORM design / Express / Koa / MySQL / MongoDB / RethinkDB / REST / GraphQL / Heroku / PostgresQL / PHP / AWS / Digital Ocean / RoR / Go

Frontend

React.js / Ember.js / Redux / jQuery / SASS / BEM / JSS / Bootstrap 3 & 4 / Apollo / Relay / Angular 2 / Vue.js

Testing

Ava / Mocha & Chai / CircleCI / TravisCI / Codecov / instanbul / nyc / Enzyme / Karma

Tools

OSX / Linux / Atom / ESLint / Babel / Sketch / Slack / GitHub / Invision / Hyperterm / Webpack / Gulp / Grunt

Principles

I try to adhere to the following principles when writing and designing software:

1. Use immutable data structures where possible
2. Stay low on the abstraction tree
3. Establish simple and deterministic data flow
4. Use small, simple, reusable, and pure functions
5. Isolate side-effects in self-contained modules
6. Separation of concerns & reusable application layers
7. Short release cycles that enable actionable feedback
8. Unit test everything & refactor often