

# Robert Polana

## Web Application Engineer

[tinyfunc.com](http://tinyfunc.com)

[robby.polana@gmail.com](mailto:robby.polana@gmail.com)

[github.com/robbywashere](https://github.com/robbywashere)

[resume.pdf](#)

[view source](#)

Debugging, refactoring, testing, grepping, console.logging, awking, acking, regularly expressioning, parsing, bashing, piping, patching, migrating, compiling, transpiling, requesting, reviewing, committing and *safely* rebasing my way through life while resisting the brazen impulse of pushing directly to prod.

### Skills

<b>Languages:</b>	Javascript	TypeScript	Ruby	Python	Go
<b>Data:</b>	PostgreSQL	MySQL	ORMs	active-record	NoSQL
<b>Deploy:</b>	docker	AWS	12-factor-apps	micro-services	serverless
<b>Style:</b>	jsx	jss	styled-components	sass	css
<b>State:</b>	GraphQL	Redux	REST	oauth + jwt + sessions	single-page-app architecture
<b>Workflow:</b>	tdd	git	tcb	testing, testing, testing	
<b>Front End:</b>	ReactJS	AngularJS	d3	semantic-ui	bootstrap
<b>Back End:</b>	NodeJS + Express	Ruby on Rails	Django		

### Experience

#### Appirio Inc.

##### Technical Lead / October 2016 - November 2017

- Mentored, tutored and evangelized new technologies and practices for a modern stack solution to meet enterprise level requirements, such as scalability, access control, auditing and security.
- Triaged created and managed issues/tasks with our offshore development team.
- Traveled on-site to meet with clients to determine project requirements and deliverables
- **Technologies used: NodeJS+express, Postgres, ReactJS+Redux, Docker**

#### MustWin

##### Lead Application Engineer Consultant / May 2015 - February 2016

- Contributed non-trivial pieces of core back-end functionality to a custom scalable Docker platform product used by developers commercially for an industry leading client.
- Designed and implemented single-page-app architecture methodology and advanced front-end components to compliment the functionality of the back-end Docker platform product.
- Showed initiative by improving testing and build integrity, which proved critical in developing a *developer platform*
- Developed from the ground up using Go, a complete back-up and restore service for CassandraDB; a NoSQL distributed database.
- **Technologies used: Go, Docker, AngularJS, CassandraDB, and Java**

## **Mimir Health Intelligence**

### **Senior Software Engineer / November 2014 - February 2015**

- Solely developed and deployed the full stack of a MVP (minimum viable product) to a big data and analytics visualization service.
- Utilized knowledge of the full stack implementation using a combination of Ruby on Rails, Angular and NodeJS technologies and their respective ecosystems to meet time-sensitive and transitory requirements during the prototyping and MVP roll-out
- Established software development cycle practices and processes - tracking of features / requirements, logging bugs, and outlining sprints
- **Technologies used: AngularJS, NodeJS, Ruby On Rails, AWS**

## **Razorfish**

### **Senior Software Engineer / June 2014 - October 2014**

- Facilitated accessibility/a11y; a key essential requirement of a particular large scale product, by implementing cross-browser knowledge, w3c compliant HTML standards and testing; including cross device and various screen reader software.
  - Supported and mentored junior developers in javascript, coffeescript, sass and css idioms and best-practices.
  - Participated and assisted in conducting open monthly accessibility meet-ups to further accessibility perspective and encourage collaboration amongst the local developer community and their end users
  - **Technologies used: AngularJS, coffeescript, sass, and css**
- 

### **ShipStation - Senior Front-End Engineer / August 2013 - December 2013**

### **Cooking Planit - Web Application Engineer / April 2012 - July 2012**

### **Spawn Labs - Web Service Engineer / August 2011 - December 2011**

### **Fuery Solutions - Junior Developer / January 2011 - July 2011**

### **Telefund - Database and Systems Administrator / March 2010 - November 2010**