

# PAUL SADAUSKAS



EMAIL  
paul@sadauskas.com



WEBSITE  
blog.theamazingrando.com



GITHUB  
github.com/paul

## > SENIOR WEB APPLICATION ARCHITECT

### Passionate about the craft of creating software

My passion is creating software that is a joy for fellow developers to both use and contribute to. I am motivated to hone my craft of software development, to learn as much as I can about better ways to accomplish complicated projects, and to build tools that are elegant, robust, and reliable.

My full-stack experiences varies from digging into the nitty gritty of kernel stack traces and debating the merits of various Javascript frameworks, and everything in between. As an engineering lead, I'm also capable of recruiting and leading a team of fellow developers and mentoring junior engineers.

## > NOTABLE PROFESSIONAL SKILLS

- Full-Stack Web Application Development
- Expert in Ruby & Rails
- Strong background in Unix/Linux Dev+Ops
- Web APIs/services, ReST & Hypermedia, JSON+LD, Hydra
- Service Oriented Architecture & Distributed systems
- System Administration & Large-scale deployments, AWS, Heroku, Docker
- Javascript/ES6+/Typescript, CSS & SASS, Elm, Ember.js, React
- Familiarity with Elixir, Rust, Go, Python, C, Crystal
- Open Source Contributor & Maintainer
- PostgreSQL, Elasticsearch, CouchDB, DynamoDB, MongoDB and other relational and non-relational databases
- Proficient at Bash scripting and Unix CLI tooling
- Writing documentation, mentoring, pair-programming; soft skills we use to hone our craft collaboratively
- Agile methodologies, Scrum

## > PROFESSIONAL ACHIEVEMENTS

Feb 2023—Apr  
2024

### STAFF SOFTWARE ENGINEER @ HouseAccount

- Architected and implemented CRM integrations system. HouseAccount needs to reliably push and pull customer data the service providers' CRMs, which are often unreliable and not built with this use-case in mind. I designed a system that allows us to re-use large amounts of logic and error handling, such that integrating with a new CRM involves implementing only a few classes and methods. In doing so, I reduced the implementation time for a new CRM from 6 months to 6 weeks, and can be accomplished by any member on the team.
- Similarly, I refactored our ViewComponents from being CRM-specific implementations to being generic, so that a single Component could be used to display the different data formats from the various CRMs. I leveraged several tools such as Presenters and Commands to accomplish this.
- Increased the interactivity of the site using Hotwire, Turbo and Stimulus. Refactored many of our Components to work more seamlessly with Turbo, and established and documented guidelines for building new Components.
- Coached and mentored the team in Rails and Web Development best-practices.
- Worked with the COO and VP of Product to define, scope and plan projects for myself and my team. Established processes between the team and product for new feature work, background chores like updates and refactoring, and prioritizing bugfixes and incident management.

Apr 2017—Dec  
2022

### LEAD SOFTWARE ARCHITECT @ TextUs

Joined as 3rd engineer on a team that had an existing MVP Rails application.

- Drastically improved reliability of message delivery by incorporating AWS technologies API Gateway, SNS & SQS, and created Terraform tooling to ensure consistency between Staging and Production.
- Improved performance and scalability of message delivery to accommodate customer usage.
- "Rescued" the existing test suite, improving runtime and reducing failure rate.
- Implemented best-practices for monitoring and instrumentation, using InfluxDB and Grafana, with Honeybadger for error notifications.

Once it became clear the existing application could not handle the scale we wished to achieve with our new enterprise customers, I spearheaded a greenfield re-write of a replacement Rails application.

- New app scaled from 1M Messages/day to over 50 Million, and 40,000 daily active users.
- Expanded use of SQS for job queues to improve application reliability and scalability.
- Implemented PostgreSQL best-practices: foreign key constraints, custom domains, materialized views. Optimized queries, index usage, and stamping down N+1s.
- Designed and built Hydra-based API for consumption by our Web UI, Mobile apps and Chrome extension, as well as 3rd-party integrators. Worked closely in conjunction with frontend engineers to develop an API that balanced request count with payload size. Championed use of Hypermedia for the team which delivered many positive benefits.
- Designed and built integration system for the reliable delivery of Webhooks and 3rd-party API calls in the presence of unreliable customer and partner services.
- Involved in design and implementation of every other major feature and component.
- Built infrastructure for deployment, instrumentation and alerting using Heroku, Terraform, InfluxDB, Grafana and Honeycomb.

In addition to the technical work, I also performed duties as Team Lead and occasionally "Acting CTO".

- Grew engineering team from 4 to 20 consisting of a diverse group of people. Streamlined onboarding, performed 1on1s with the team.
- I strongly believe in hiring junior engineers and continuing their education. About 1/3 of our engineers were recent graduates of boot camps, often as their first engineering job. Closely mentoring and providing a supportive environment is a key part of this process.
- Created the technical interview process, which was used to hire most of the team. I performed most of the early interviews, then trained the team to also be able to conduct interviews, and iterate on the interview process.
- Defined features, scoped product objectives, and collaborated with stakeholders to build a company roadmap.
- Instilled a culture of positivity and supportiveness, so the team felt comfortable experimenting and learning from each other.
- Championed and supported Agile methodologies for the team, leading standups and retros. Strong advocate of introspection and continuous improvement of every aspect of building software and teams.

Nov 2021

#### SERVICE OBJECTS WITH DRY.RB: MONADS AND TRANSACTIONS @ RubyConf Denver

*This is a more polished version of the talk I gave at Boulder Ruby.*

Service objects are an important tool in your toolbox, and Dry.rb's Transaction library is one of the most powerful, and one of the most magic. It's a "business transaction" DSL, and has error handling as a primary concern. We'll start by exploring Monads in Ruby (they're not scary!). Then we'll see how that simple concept unlocks another level of service objects that are far more robust and testable, and how to wire them all together with Dry::Transaction. Finally we'll touch on extending transactions with custom steps, and how to integrate them into an existing application.

Jun 2020

#### IT { SHOULD BE A\_GOOD\_TALK }; WRITING CUSTOM RSPEC MATCHERS @ Boulder Ruby Group

Writing custom RSpec matchers can be helpful for writing simpler, DRY-er and more descriptive specs. We'll talk about how they're used, how to make them, and some useful ones I've discovered or created myself.

I live-coded this talk, and there was a great discussion afterwards where it was helpful having that coding environment available to show participants.

Apr 2020

#### BETTER SERVICE OBJECTS WITH DRY::TRANSACTION @ Boulder Ruby Group

Service objects are an important tool in your Rails toolbox, and dry-transaction is one of the most powerful, and one of the most complicated. It's a "business transaction" DSL, and has error handling as a primary concern. This talk will give a background of how to use it, how to integrate it with your Rails application, and advanced usage with some custom steps.

Feb 2015—Mar  
2017

#### SENIOR ENGINEER @ Kapost

- Architected and implemented an application to collect and present analytics data. Utilized CouchDB and eventually Elasticsearch for analytics storage and aggregation, and SQS for background workers.
- Transitioned to a DevOps role, assisting other teams in building out multiple apps in a microservices architecture.
- Built scripts and configs to create a Docker-based development environment to simplify developing against other teams' apps.
- Collaborated with teams to develop their features and products in a way that integrated well with existing projects.
- Built Chat-Ops bots to allow team members to inquire about the state of the system, and deploy & provision ad-hoc staging environments.

Aug 2014—Feb  
2015

#### SENIOR ENGINEER @ RainforestQA

- Wrote a full-stack Rails application (Postgres, Ruby, CSS, JS) to provide testers with a one-time use email address
- Added features to the open-source QueueClassic and QueueClassicAdmin gems
- Troubleshooted performance issues relating to load from a "stampeding herd" effect from testers
- Prototyped a replacement for a core component of the UI in Ember.js

Dec 2012—Aug  
2014

#### SENIOR ENGINEER @ Librato

- Architected and implemented a queue/worker system in Ruby to import users' AWS Cloudwatch data into Librato Metrics
- Contributed to and coordinated feature development on the Librato open-source client libraries
- Drove greater adoption of hypermedia concepts in the Librato API, both external and internal
- Took charge of a website security audit and fixes to user-facing websites and APIs
- Evaluated various JavaScript frameworks to use in a new frontend dashboard

Jun 2012—Nov  
2012

#### API DEVELOPER @ GitHub

- Design and implemented hypermedia APIs for extracted micro-apps to communicate with the main app and each other
- Made upgrades to various aspects of the GitHub API, and communicated those changes to users
- Improved tooling for generation of documentation of the GitHub API

Nov 2011—Jun  
2012

#### SENIOR SOFTWARE ENGINEER @ LivingSocial

- Took part in upgrades and enhancements to the email newsletter and notification delivery system
- As a member of the API team, drove adoption of hypermedia APIs as part of a SOA refactor
- Helped refactor a large Rails app by following the principles of TDD and Martin Fowler's *Refactoring*

May 2011—Oct  
2011

#### SENIOR SOFTWARE ENGINEER @ Strobe

- Implemented a "ChatOps" continuous deployment system using Campfire and Chef
- Created from scratch the Chef cookbooks and recipes to deploy the entire Strobe infrastructure to AWS
- Set up a JenkinsCI cluster for continuous builds of Ruby/Rails, C++ and Clojure applications
- Assisted other team members in getting their projects in CI and Chef for deployment
- Helped implement a CDN/Caching proxy for customer assets in Clojure and Riak

Oct 2010

#### FORMS DON'T HAVE TO BE THIS COMPLICATED @ Mountain.rb

In this talk, I lament how difficult it is to build complicated nested form in Rails & UJS. I offer a few solutions, including Form Objects. More recently, pure front-end apps in React or Ember have supplanted the need for a comprehensive solution in Rails itself.

Mar 2010

#### HOW HTTP ALREADY SOLVED ALL YOUR PERFORMANCE PROBLEMS 20 YEARS AGO @ MountainWest RubyConf

I describe several of the more powerful features provided by the HTTP protocol, which are supported by most basic HTTP clients in languages' stdlib, but not taken advantage of by most advanced client libraries.

Mar 2009

#### WRITING ADAPTERS FOR DATAMAPPER @ MountainWest RubyConf

As a member of the core team, I gave a tutorial on how to write custom adapters for the DataMapper ORM. Several adapters for other databases and web services were contributed as a result of this talk.

Apr  
2011—Present

#### PROGRESS\_BAR @ Creator

- A very simple implementation of a CLI progress bar, similar to the output of cURL
- Mentored contributors to keep their pull requests small, focused, and well-tested
- It was originally intended primarily for personal use, but is now a dependency of several popular projects

Oct 2007—May  
2011

#### RESOURCEFUL @ Co-Creator

- Implementation of a fully RFC2616-compliant HTTP/1.1 client in Ruby

- Includes support for advanced features of HTTP, including support for redirects and caching with pluggable backends
- Used to build other abstractions, such as object mappers for REST/Hypermedia APIs

2008—2010

</> **DATAMAPPER** @ Core Team

- Heavily involved in design and implementation of DataMapper, and preparing for release of 1.0
- Responded to issues and pull requests, and mentored new developers to get involved
- Redesigned and implemented DataMapper Adapters interface
- Designed and implemented Migrations plugin

2008—2010

</> **MERB** @ Contributor

- Heavily involved in design and implementation of Merb, preparing for release of 1.0
- Implemented support for conditional GET requests (304 response/Etag)
- Implemented correct handling of HTTP Accept header including negotiating client and server priorities

2009—2010

</> **COLLECTD** @ Contributor

- Contributed the tokyo-tyrant collection plugin
- Contributed the write-http plugin, which will submit collected measurements via libcurl to an HTTP server

2006—2011

👤 **SENIOR SOFTWARE ENGINEER** @ Absolute Performance

- Rewrote an existing monolithic Perl web application into multiple Rails applications
- Architected and developed new applications to interact with the main apps by hypermedia APIs
- Accomplished an order of magnitude improvement in capability by implementation of HTTP caching
- Revamped procedures to ship and deploy applications in various languages to remote customer sites
- Implemented new features in various NoSQL databases, TokyoTyrant, MongoDB, and Redis
- Encouraged the team to adopt automated Unit and Functional testing, as well as agile methods
- Engaged in continuous deployment on non-cloud physical hardware at various customer locations
- Involved as primary architect of all layers of the stack, from Javascript UIs to database schemas

2002—2006

👤 **SENIOR WEB DEVELOPER** @ InputOutput Unlimited

- On a contract basis, implemented numerous web applications in PHP and later in Rails
- Created specifications and designs from (sometimes ambiguous) customer and stakeholder requirements

## DOWNLOAD RESUME

[Markdown/txt](#)   [PDF](#)   [Source](#)