




PAUL SADAUSKAS

›SENIOR WEB APPLICATION ARCHITECT

 EMAIL
paul@sadauskas.com

 WEBSITE
blog.theamazingrando.com

 GITHUB
github.com/paul

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+, CSS & SASS, Elm, Ember.js, React
Familiarity with Rust, Go, Python, C, C++, Crystal
Open Source Contributor & Maintainer
PostgreSQL, Elasticsearch, CouchDB, Riak, 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

›CONFERENCE TALKS

- Service Objects With Dry::rb: Monads and Transactions
RubyConf Denver - Nov 2021

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.

This is a more polished version of the talk I gave at Boulder Ruby.
- It { should be a good talk } : Writing Custom RSpec Matchers
Boulder Ruby Group - Jun 2020

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.
- Better Service Objects with Dry::Transaction
Boulder Ruby Group - Apr 2020

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.
- Forms Don't Have to be this Complicated
Mountain.rb - Oct 2010

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.
- How HTTP Already Solved All Your Performance Problems 20 Years Ago
MountainWest RubyConf - Mar 2010

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.
- Writing Adapters for DataMapper
MountainWest RubyConf - Mar 2009

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.

›OPEN SOURCE PROJECTS

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

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

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

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

Collectd

Contributor

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

Additionally, I have made smaller code contributions to numerous other OSS projects, including Rails, Rack, Awesomewm, KDE, and Arch Linux. I also tinker with smaller projects frequently and publish them to [GitHub](#).

»PROFESSIONAL EXPERIENCE

LEAD SOFTWARE ARCHITECT @ TextUs Apr 2017 - Present

- Drastically increased reliability and performance of existing MVP application by implementing RDBMS best practices and background queues.
- Led development and architecture of greenfield rewrite of the application, once it became clear the MVP in place was insufficient for our new larger customer profile:
 - Designed and built infrastructure to deliver and receive 500MM SMS messages a year across multiple providers with 100% reliability, utilizing AWS Products and integrating them seamlessly with Rails' job backend.
 - 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 in the presence of unreliable customer and partner services.
 - Involved in design and implementation of every other major system and component.
- Built infrastructure for deployment, instrumentation and alerting using Heroku, Terraform, InfluxDB and Grafana.
- Partnered with CTO to scope product objectives, define features and manage the engineering team.
- Grew engineering team from 4 to 20 consisting of a diverse group of people. Streamlined onboarding process for mid- and senior-level engineers, closely mentored junior engineers.
- Instilled a culture of "its ok to make mistakes", 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.

SENIOR ENGINEER @ Kapost Feb 2015 - Mar 2017

- 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.

SENIOR ENGINEER @ RainforestQA Aug 2014 - Feb 2015

- 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 "stamping herd" effect from testers
- Prototyped a replacement for a core component of the UI in Ember.js

SENIOR ENGINEER @ Librato Dec 2012 - Aug 2014

- 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

API DEVELOPER @ GitHub Jun 2012 - Nov 2012

- 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

SENIOR SOFTWARE ENGINEER @ LivingSocial Nov 2011 - Jun 2012

- 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*

SENIOR SOFTWARE ENGINEER @ Strobe May 2011 - Oct 2011

- 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

SENIOR SOFTWARE ENGINEER @ Absolute Performance 2006 - May 2011

- 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

SENIOR WEB DEVELOPER @ InputOutput Unlimited 2002 - 2006

- 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

»TECH I'D LOVE TO USE & LEARN MORE ABOUT AT MY DAY JOB

Rust
Crystal
Elm
Reverse Engineering
Hardware Development

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