

DANIEL P. CLARK

RUBY EXPERT, RAILS, AND RUST DEVELOPER

6ftdan@gmail.com 
github.com/danielclark 

Who am I, or more importantly, who am I as is relevant to you. I'm a dog with a bone when it comes to problem solving. My drive for problem solving is fierce and I thoroughly enjoy it. My ability to express ideas in such a way that helps the listener understand new concepts is well honed. I'm cordial in my communication and very much enjoy communicating with people. But more importantly I'm a good listener with keen perception toward intent on what's being said. I'm well practiced in writing, documenting, and providing educational material. And I have a keen focus on producing quality code by following what's idiomatic for each particular language, best practices in design patterns, thorough testing by both unit and integration tests, and a love for writing performant code verified with benchmarks. I'm a polyglot always learning new languages and technologies and am quick on the uptake.

My background; I was first introduced to programming as a child by my father. I later started programming in 2001 as a hobby of passion. In 2008 I delved into some remote freelance work and that went well. Then as of 2014 I transitioned to developing software full time and writing articles on personal development, teaching programming, website development, and operating system management and troubleshooting.

If I could have my dream team environment it would be one that is friendly, open, collaborative, peace-driven, and trusting as a community of highly motivated software development peers. With opportunities to mentor as equals and bond as those who share in a common goal. With a focus on quality rather than quantity for code practices to produce superior systems that can stand the test of time. Where good practices such as *Red, Green, Refactor* are the expected discipline to follow and peer review is handled with grace and timeliness. Where all individuals are treated with respect and the language we speak is carefully crafted to be politely diplomatic even when ideas or code is to be challenged. Where to be winsome is valued preeminently above any dismissiveness of ideas, code, or values. Where people come before code.

Highlighted Skill Experience

Authored [FasterPath](#)

- With this projects first release I made the pages of my Rails site load more than 30% faster. I discovered which methods took up the majority of the time for a Rails web page to load and rewrote one specific method to accomplish this. Since then my project has 15 such methods rewritten from Ruby into Rust that all now run faster. This project is what I'm most known for.

Ruby — since 2012

- I am an expert in Ruby from design patterns, best practices, to the underlying behavior. I've also gained much experience in implementing Rust code to interface directly with Ruby's C code. I've worked with many APIs and created abstraction layers to simplify their use.

Rails

- Initially I was deeply involved in Rails work as the sole full-stack developer of a start up company for over 2 years. From this I have gained expertise in many areas from which I've written many blog posts teaching front end techniques & technologies such as UJS, AJAX, WebSockets, and ActionCable, to backend best practices.
- I'm well experienced with CSS, SASS, CoffeeScript, JavaScript, asset management, and templating systems.
- I've built and deployed containerized websites with Docker and Heroku with various customizations.
- Some Rails projects I have had experience in:
 - Warehouse inventory system with book & video support
 - A YouTube video research tool for efficiently finding sports games
 - A Podcast aggregator with web player & trends ranker
 - A VueJS centric commute cost calculator
 - A connections management system with high integration to social networks and focussed on following up with conversations
 - An educational competition platform for solving puzzles in a language of your choosing
 - An animal & insect species catalog & tracker system
 - And a few that didn't make it to being deployed such as a online game club, a project performance statistics cataloger, and a definition/context driven blog enhancemer to clarify meanings

Rust — since 2015

- I've over 3 years experience in Rust with a focus on writing performant and idiomatic code. This includes the project FasterPath with plenty of benchmarking and tracking data.
- I've good experience with multi-threading in Rust with my project [Abrute](#) which utilizes all CPU cores to attempt passing file encryption via the brute force method. Starting from the ground up I wrote a character sequencer which implements many algorithms, the frontend client experience featuring several command line statistical report systems, and a backend JSON endpoint for cluster data aggregation. Abrute is built with server clustering in mind allowing splitting work loads across any number of systems.
- Foreign function interfacing; with my experience with FasterPath, and Rutie, I've become accustomed to reading C source code to write Rust interfaces around and allow Rust and C to call each others code.
- Wrote a tool to allow the JavaScript ecosystem to be available with Rust and deployed a minimal example using the Rust framework [Gotham](#) with VueJS and CoffeeScript for the front end. [\[read article\]](#)

JavaScript

I have adequate experience with the following JavaScript technologies:

- Vanilla JavaScript (*standard JavaScript, not the Vanilla JS library*)
- CoffeeScript
- VueJS
- UnderscoreJS
- jQuery
- AJAX
- Turbolinks

Docker

I have experience designing and building many docker containers for

- **Continuous Integration** — test suites & CI project compilation and releases through Github Releases
- **Build Staging** — compilation and code execution of many programming languages each in their own Docker container as a stage execution test suite.
- **Web Deployment** — custom Docker containers designed to work the same both on Heroku and the local system.
- **Development Environment** — with [ruby-pair](#) I've built my ideal pair-programming environment which allows remote pairing and development with several programming languages.

Linux

- Around 20 years experience working with Linux. Written many blogs on Linux system applications, building, configuration, management, and backup. Very capable in shell scripting & system administration.
- 1 year experience with [ProxMox Virtual Environment](#); an open-source server virtualization environment. Which allows deployment and management of virtual machines and containers.
- Experience customizing Linux installation images. Used various medium options: Virtual Machine, USB, CD, Hard Drive Bootloader. And created a Linux installation server which allowed Network Booting systems to install the Linux OS over a LAN cable.

APIs

- Implemented **OmniAuth** with **Evernote**, **Twitter**, **LinkedIn**, **Google**, **Facebook**, **Yahoo**, and **Meetup.com**
- I've written an API client library for **Travis CI** called [Trav3](#)
- I've written an application which utilized the **Twilio** API
- I've written an application which utilized the **Evernote** API and specifically its OCR feature to gather data from images. I've also written a blog on how one may integrate their Rails app with it titled: [Working with the Evernote API in Rails](#)
- I've written application integration for **Google Contacts**

Technical Skills

- **Proficient in:** Rust (Since Oct 2015), Ruby (Since early 2012), Ruby on Rails, TDD, Capybara, Docker, REST, APIs (Github, Travis CI, Twitter, Evernote, Twilio, etc.), Git, Crystal, JavaScript, jQuery, CoffeeScript, CSS, Gem Authoring, HTML5, JSON, Linux SysAdmin, Windows SysAdmin, Unix Shell Scripting, Computer Technician, VIM
- **Web Automation with:** Continuous Integration, Mechanize, Nokogiri, Selenium, and Watir
- **Moderate experience in:** ActionCable, Agile, AJAX, Assembly, Bash, Batch, CGI, CMS, Django, Encryption, Excel, FFI, FishShell, L^AT_EX, Metaprogramming, OpenSSL, PostgreSQL, Python (Since 2001), RSS, Satchmo, SEO, SOLID, XML, Security Practices, VueJS, Ruby C API
- **Studying:** Algorithms, CoffeeScript, Design Patterns, FP, L^AT_EX, Ruby C API, Rust, VueJS

Additional Experience Highlights

Professional Ruby & Rails Blogger

- I've been writing educational blogs for many years at 6ftdan.com . One of the companies I freelance for in this regard is Codeship.com . I write about technologies, programming languages, and personal development topics.

Ruby for Good

- **2016:** Charity Rails development for Virginia Working Landscapes [\[link\]](#). I joined a team of developers in providing free software development for helping catalog living things in nature. This information can then be used to help determine issues and be of help to protect any living species that may be endangered.
- **2017:** Charity Rails development for Loudoun Codes [\[link\]](#). I joined a team of about 10 developers in building a Rails project for high school programming competitions. The system is responsible for running students code submission, grading the results with time and penalty rankings, as well as an official scoreboard system. My primary role in this project involved implementing Docker client runner implementations for various programming languages which includes Ruby, Python, C++, Rust, Haskell, and Crystal. This was refactored into the primary implementation work done by Chris Hoffman.

An Undisclosed Start-up

- Full stack Rails developer and CTO. Rails, Ruby, AJAX, JavaScript, jQuery, APIs, OAuth, OmniAuth, Devise, ActiveAdmin, Heroku, and much more.

The Minitest Cookbook — *Technical Proof Reader*

- Chris Kottom authored the book “[The Minitest Cookbook](#)” — with three weeks to go until it was due to be published I read through the book twice giving editing advice on each iteration. In 2017 I've also gave a final proof read for Christ Kottom's “[Developing Rails API Applications](#)” book.

RailsRumble

- **2014:** I joined this project with a team assembled of 3 strangers from Twitter. As part of a code competition I built an interface for scraping YouTube videos and wrote some helper methods to take advantage of creating embedded website videos. One of the core developers for that Rails project.
- **2015:** I started this project with, again, a team of strangers on assembled via Twitter. In 48 hours we built TrenderCast which is a podcast subscription and trends system. This product was my vision and I was the primary Rails developer on this project.
- **2016:** This year I focussed on working solo and experimenting with new Rails technology of WebSockets/ActionCable. The project was to build a game server for poker games. With the 48 hours I made it as far as implementing a chat room, game tables, seat assignment, and card dealing.

NetVersa ~ 2012

- Designed full website automation for popular sites using Watir and Watir-webdriver in the Ruby language.

Scholar Square — 2008

- Designed a script to pay all site members who've authored content; all appropriate fees for students memberships, and subscriptions, via the PayPal API using Python.

eyePhoneRevival

- This was a profitable startup for refurbishing and selling iPhone and surplus part sales. I designed and deployed a full eCommerce website built on top of the Satchmo/Django framework using Python. *Note the profits were made before the site was and the site was a dud.*

HSPN

- Designed a basketball tournament ranking application which ranked teams on many different criteria. Designed my own flat file database system to keep records in CSV. This was used to rank a multi-day interstate basketball tournament.

Open Source Contributions

- Charity Rails development in [Ruby for Good](#) for Virginia Working Landscapes [\[link\]](#) and Loudoun Codes [\[link\]](#).
- Minor contributions to several major Open Source projects including Ruby, Rails, Crystal, RubyGems, RVM, Bundler, and OmniContacts
- Continued development of [Ruru](#) with an official fork [Rutie](#) — A library to allow you to write more performant code for Ruby in Rust and vice versa.

- Authored [language_cards](#) — A command line flash card game, written in Ruby, with it's own internal MVC architecture, ERB templates, I18n support, and YAML integration for translations.
- Authored [webpacker-cli](#) — Makes the features of [Webpack](#) and Rails' convenience of [Webpacker](#) available for and language/framework via the command line.
- Authored [PolyBelongsTo](#) – An advanced Rails – ActiveRecord relationship library.
- Authored and contributed to around 80 other Open Source libraries available on [Github](#).
- Open Source game released in 2002 called [Py Connect Four](#) written in Python using the PyGame SDL library

Technical Blog Post Headlines

- [Software Development: The Road to Mastery](#)
- [Gotham — From Start to Heroku](#)
- [What's the Difference Between Implicit vs. Explicit Programming?](#)
- [Apply Changelog Best Practices to Development](#)
- [Familiarize Yourself with Ruby String Encoding](#)
- [Measuring Performance Metrics in Rails](#)
- [Mathematics with MathJS](#)
- [Rails Frontend Testing with JavaScript Insights](#)
- [Refactoring for the Tell Don't Ask Pattern](#)
- [VueJS Components with CoffeeScript for Rails](#)
- [VueJS as a Frontend for Rails](#)
- [Micro Backups of Your Git Dev Work](#)
- [Understanding Rust Loops](#)
- [ProxMox; A VM Server For Your Home](#)
- [Rust Design Considerations with Borrowing](#)
- [Mixing Ruby and Rust on Heroku](#)
- [Improving Ruby Performance with Rust](#)
- [Creating Configuration Objects in Ruby](#)
- [Understanding Lifetimes in Rust](#)
- [Let's Talk About Shell Scripting](#)
- [Elixir Envy ▷ Ruby](#)
- [Advanced Enumeration with Ruby](#)
- [Graphing Benchmark Results in Ruby](#)
- [Refactoring and Design Patterns](#)
- [Crystal from a Rubyist's Perspective](#)
- [ActionCable: The Missing Guide](#)
- [Creating Powerful Command Line Tools in Ruby](#)
- [Leveling up with Git](#)
- [Working with the Evernote API in Rails](#)
- [Creating Advanced ActiveRecord DB Queries with Arel](#)
- [Get Started Quickly with Docker and Sidekiq](#)
- [Building a Well-Polished Ruby Gem](#)
- [Re: What is the splat operator doing here?](#)
- [Getting Started with Minitest](#)
- [Managing Private Dependencies with Bundler](#)
- [The Pros and Cons of Ruby Refinements](#)
- [How to Inspire Community in Programming](#)
- [Visualizing Algorithms Before Implementation](#)
- [Coming to Rust from Ruby](#)
- [Unobtrusive JavaScript via AJAX in Rails](#)
- [A Deep Dive into Ruby Scopes](#)
- [How to Build the Ultimate Remote Pair Programming Environment](#)
- [Encrypted Linux Backup with Google Drive and Duplicity](#)
- [Use Ruby's Refinements Anywhere With An Anonymous Class](#)
- [Rails: Has One Through Polymorphic Relation](#)
- [Ruby: Bindings Across Inheritance](#)
- [101 Ruby Code Factoids](#)
- [Ruby: Arrays by Example](#)
- [Iterative Evaluation](#)
- [Implement a Lazy Hash in Ruby](#)
- [Rails Rumble – the Experience](#)
- [Introspection by Design in Ruby](#)
- [Install Ubuntu 15 Over the Network PXE/BootP/TFTP NetBoot](#)
- [Ruby Refinements – Not quite ready for production](#)
- [Rails' HashWithIndifferentAccess](#)
- [Switching From Unicorn to Puma on Heroku is Easy!](#)

- Speed Up Your Rails Views By Deferring Changes
- Rails: Don't "pluck" Unnecessarily
- Fake a SMTP Server with Python for Rails Testing
- HABTM; You Don't Need It
- Chaining an Array into Arel ors
- Private Module Methods in Ruby
- Have problems testing methods that query ActiveRecord?
- Documenting Your Code
- ActiveRecord vs Me... Round 1 – FIGHT!
- Getting Started with MiniTest
- Rails 4's Awesome enums
- Different Collection Types in Ruby
- Hooks
- More UJS fun with Accordion Content
- Substitution with Regexp Groupings
- Ruby: Actionable Meta
- Configuration With a Singleton Instance
- Numeric Bases and Defining Your Own
- Ruby: The Case for Case
- Version Both Your Github Repo and Gem
- Value Assignment and Deferment with Lambdas/Procs
- Writing methods for both class and instance levels
- Minitest with Watir-webdriver
- Some Basic Ruby Tools for Sniffing out Errors
- A Glimpse at Tail Call Recursion
- Use the given project generators
- Setting up Remote SSH Access to your Ubuntu Box
- Rails Polymorphic Associations
- Do It All With Ruby
- Getting Into Rails Model Queries
- Using Ruby Object Type Classes to Safely Build Data
- Discovering UJS with AJAX
- Rails Helper Methods
- or
- Learning Your First Programming Language
- Refinements over Monkey-patching
- The 500 Millisecond Rails Partial
- Email Composer/Editor Template for WebDevs
- Vanilla JavaScript – Stepping into the language
- Encryption in Ruby on Rails
- JavaScript is easy with jQuery
- Dealing with Rails 4 forms and params
- When to use Ampersand and/or Colon with Proc, Map, and Inject
- "Assume Everything Will Break" Development
- Free Travis CI for your Open Source project
- Ruby & Ruby on Rails Methods for Discovery Part 1
- Create a Game Instance in Rails
- Manual Polymorphic Creation in Rails

Spoken Languages

US English (Native)	● ● ● ● ●
Spanish (studying)	● ○ ○ ○ ○
Japanese (studying)	● ○ ○ ○ ○

Education

Clark Christian High School 2003