Ron Gruesbeck

Web Developer
Technologist

Email: ron@rongruesbeck.com
Homepage: http://rongruesbeck.com/
Github: https://github.com/rgruesbeck/
Last Updated: May 31, 2017

Summary

I am a Web Developer and Technologist with 4 years of web development experience, 5 years of programming experience, and 14 years of experience using the command-line & shell scripting on GNU/Linux. My technology exposure includes JavaScript, Ruby, Python, Go, R, Docker, AWS, React, Angular, Express, Ruby on Rails, HTML, and CSS.

I am partial to Unix style programming, and Unix culture in general. I have found the Unix Philosophy of modularity and reusablity to be a practical design asthetic that produces maintainable, reusable, and flexible software that adds value, reduces costs, and is enjoyable to work with.

My personal projects include a Web RCT signal server and client for browser to browser file transfer, and a small web interface for searching and viewing Jazz leadsheets hosted by the Internet Archive.

Skills

Web Development

Web Clients: I have built everything from static websites to web apps. In keeping with Unix Philosophy, I prefer to compose solutions using NPM packages with a clear purpose, although I have experience with frameworks including React and Angular 1. I am familiar with the common design patterns used in web apps including Pub/Sub patterns for managing application state and the more data centric MVC pattern. I am also familiar with the common development tools: Git, NPM, Gulp, Browserify, & WebPack.
Web Servers: I have built and deployed web-servers with NodeJS, Ruby on Rails, and Go. From this I have gained experience in the following areas. NodeJS: (WebSockets, LevelDB, MongoDB, RPC with dnode, Express, JSON Rest API.) Ruby on Rails: (Rails, PostgreSQL, and Caching.) Go: (Gin, Context Object Pattern, Caching)

I have also have experience maintaining existing projects.

Testing

I like writing test with tools that write TAP output. Using this convention provides flexible reporting and increased sensitivity to false positives. I am familiar with RSpec for Ruby on Rails, and Tape for NodeJS and Browsers, and use Testling for cross-browser testing.

Systems Administration

My main development environment is Arch Linux on ThinkPad Laptops for local development, but prefer Ubuntu for Docker images and when using EC2 or Digital Ocean. I move all medium-sized and larger projects to Docker containers. My editor of choice is Emacs with in Evil mode.

Documentation

I like writing documentation in open plain-text formats like Markdown, LaTex, or Org files. This makes collaborating, versioning, and searching easy, while HTML and PDF versions can be created quickly with generators.

Recent Projects

Arts Council Napa Valley - Development, Maintenance

Ruby Developer; August 2013 - Present; remote, part-time

Skills used: Ruby, Ruby on Rails, PostgreSQL, Docker, JavaScript, HTML, CSS.

Role overview:

- Full Stack Developer on website for publishing news, events, artist profiles using an admin interface with WYSIWYG support. In addition to implementing new features, I am responsible for maintaining server reliability, updating and refactoring the codebase, and ensuring good test coverage. I also perform upkeep tasks such as version control with git, and migrating the development environment to Docker to improve portability.

Interesting challenges:

- The client wanted to write copy for the site with a WYSIWYG. I originally had setup copycopter for them, but when they asked to be able to edit from the admin interface and use a WYSIWYG with more features I implemented a drop in replacement.
- The client needed to pull XML from a slow external source. I cached the external XML as a Ruby Hash using Rails' low-level caching feature with an expiration of 24 hours. The route would then serve up a JSON version of the data from the cache. This improved performance and limited connecting to the external source to once per day.

JubileeUSA - Consulting, Support

Webmaster; March 2016 - Present; remote, part-time

Skills used: Bash, AWS:S3, AWS:Route53.

Role overview:

- Consulting and support during transition away from unstable legacy system and recovery from malware. I setup a static mirror of the site with an S3 bucket, fallback route with Route53, and shell scripts to sync updated content to the static mirror.

Spaceblox - Consulting, Prototyping

Consultant; Jun 2016 - April 2017; remote, part-time

Skills used: Javascript, NodeJS, Go, React, AWS:S3, AWS:Lambda, HTML, CSS.

Role overview:

- Designing and implementing prototype of web app for space and facilities management. The client prototype was written in React, and a web-server was written in Go. I also prototyped the SVG conversion service for deployment as a micro-service on AWS:Lambda.

Interesting challenges:

- The client required loading SVG files containing a map of a building floor into the browser quickly. I implemented a service to minify, gzip, and base64 encoded each SVG file for storage in the associated JSON object. On the client I implemented the reverse process before loading the SVG in the browser. The process was implemented on both ends in a streaming style.
- After refactoring routes and models into their corresponding pairs of modules, I needed a way to cleanly pass in database and cache access to modules. I found the Context Object Pattern to be a clean and flexible way to provide this access with the advantages of instantiating the corresponding interfaces only once per server run, and without using global variables.

Eldorado Arts Council - Development, Stripe Integration

Javascript Developer; August 2014 - October 2015; remote, part-time

Skills used: Javascript, NodeJS, Express, MongoDB, Angular 1, AWS:S3, Stripe, HTML, CSS.

Role overview:

- Develop website and admin interface for publishing news, events, and later on processing of donations and ticket sales. The system included 2 Angular 1 clients, with one serving as the admin interface, and 1 server written in Express. Aside from the UI, I was responsible for the design, implementation, and deployment of the system.

Interesting challenges:

- The client later added the requirement of processing payments for tickets on their own. I implemented the payment process with the Stripe API that included the UI for entry and validation of payment information, confirmation and sending of email receipts.

Education

Pacific Union College Biology, BSc; 2010.

South Dakota School of Mines & Technology Introduction to C++; 2007.

GitHub

WebRTC File Transfer Prototype of a service to facilitate browser to browser file transfers. Repositories: webrtc-brower-client. webrtc-signaling-server.
Jazz Leadsheet Search Interface for searching for jazz leadsheets hosted by the Internet Archive. Repository: leadsheet-search. Link: leadsheet-search.

Languages

English (native).