Aaron Schrab

Durham, NC 27703
aaron@schrab.com
https://schrab.com/aaron/resume/

Summary

Over twenty years of experience in developing applications for the web environment. Most recently focused on backend development and deployment automation. I have experience over the full stack and try to keep up with developments in front-end technologies, and I am able to quickly get up to speed on new platforms.

Able to work both on my own or with a team. I enjoy mentoring more junior developers as well as learning new things from others.

Great attention to detail and a track record of diagnosing and resolving complex problems.

Skills

Programming Languages: Ruby, TypeScript, JavaScript, Java, Python, SQL, Perl, Bourne shell, C, C++, TCL, Expect, awk

Tools: Ruby on Rails, Node.js, React, Jenkins, PostgreSQL, git, Docker, AWS CloudFormation, AWS Lambda, AWS API Gateway, Webpack, Angular, Catalyst, Subversion, ExtJS, Asterisk, Apache web server, BIND, CVS, make, gdb, m4, MySQL, Squid, postfix

Links

- schrab.com/aaron
- github.com/aschrab
- linkedin.com/in/aaronschrab
- stackoverflow.com/users/1507392/qqx

Experience

Files.com Backend Engineer
February 2024 – January 2025

Implemented new features, diagnosed bugs, and provided fixes within the Files.com API. This is a Ruby-on-Rails based project using Amazon Aurora MySQL, Redis, and Elasticsearch for storing and retrieving file metadata. Features that I developed included support for a new read-only administrator role, and the ability to automate retrieval and saving of HTTP resources.

Worked on generation of SDKs and associated documentation for the Files.com service. One major contribution I made to this was to generate most of the content for README files from content written for the developer documentation site.

Worked to convert data stored in Redis from using JSON to MessagePack, using a combination of Ruby, Java, and Lua.

NetApp API Engineer
Morrisville, NC November 2015 – April 2023

Worked to move business rules from monolithic, client-side procedures into more structured code in API server developed with Ruby on Rails.

Mentored team members who were new to Ruby and Ruby on Rails.

Took part in overseeing the development of the beginnings of a new UI developed with Angular.

Converted UI build system from grunt to webpack, eliminating the need for bower which was not supported in the target environment.

Developed API for sending push notifications to mobile devices, using Amazon SNS.

Wrote Hive queries to gather data from a Hadoop cluster. Wrote a Python script to monitor NFS file store for updates and load this data into a MongoDB database. Developed APIs to serve the data using Node.js and Express. Developed UIs to display the data using Angular.

Deployed services to run in Docker containers on Amazon EC2.

Developed serverless APIs to operate as AWS Lambdas via the AWS API Gateway, using both Node.js and Java. Automated deployment of these using Jenkins and AWS CloudFormation.

Documented APIs with OpenAPI specifications, and deployed Swagger UI for viewing that documentation.

Created a React-based app for testing OAuth login process.

Worked on GraphQL API server developed in Node.js using the Apollo framework to work with data stored in a MongoDB Atlas database as well as working with other backend-APIs. This was then packaged in a Docker container and deployed to AWS Elastic Container Service.

Sunbird Software Senior Software Engineer
Raleigh, NC May 2015 – October 2015

Worked on the PowerIQ web interface which uses Ruby on Rails to present data from a PostgreSQL database.

Optimized queries to the PostgreSQL database in order to decrease the amount of time users need to wait to see data.

Built a Rails engine to serve JSON data and JavaScript code to build SVG charts replacing a legacy, Flash-based solution.

Wrote database migration rules using Liquibase.

Reviewed GitHub pull requests from other team members.

NetApp Tools Developer
Morrisville, NC September 2013 – February 2015

Development on the internal issue-tracking system to convert it to using an Oracle database via a JSON API implemented with Ruby on Rails.

Worked on generating testable API documentation to ensure that the described API was accurate and would remain so as additional development was done.

Helped in design of database schema.

Developed a Perl library to minimize changes needed for legacy client applications to work with the new storage system.

Participated in code review process for changes by other team members to improve code quality.

Developed tools to compare data in Oracle database against legacy storage to check for bugs in the data conversion process.

Developed Oracle PL/SQL stored procedures to assist in writing queries to replicate functionality from legacy system.

Assisted users in writing SQL queries based on their described requirements.

Corvisa Founding Engineer
Milwaukee, WI September 2009 – May 2012

Full-stack development on the initial version of a web-based application for mortgage companies using Perl's Catalyst MVC framework and PostgreSQL to run the backend which provided a REST API for a browser-based user interface implemented with the ExtJS toolkit. The initial development to assist with regulatory compliance in dealing with appraisals was done largely by myself. As the scope of the product grew to handle more functionality the size of the team grew to more than 20 developers.

Developed Perl-based tools for keeping database schema and base data synchronized with changes kept in a version control system.

Developed a system for documenting REST-API endpoints which generated PDF documents that gave examples of data structures using either XML or JSON syntax from a common XML source format.

Enhanced WireShark's support for dissecting AMQP messages and added support for binding a queue to an exchange based on message headers to the Net::AMQP::RabbitMQ perl module.

Managed the migration of the source code from subversion to git.

Brevient Technologies Senior Software Architect
Milwaukee, WI January 2007 – September 2009

Developed a web-based system for creating marketing messages and distributing them via email, RSS, and website applets using Perl's Catalyst MVC framework along with PostgreSQL.

Used Perl to implement a system for billing thousands of VoIP calls per hour along with other products, generating PDF invoices that could contain more than 10,000 pages each.

Maintained the Jyngle.com site using Perl's Catalyst framework, which allowed users to send voice and/or text messages to large groups of people at a time.

Enforcer Group, Inc. Senior Developer
West Allis, WI November 2002 – December 2006

Led the design and development of web-based records management, mugshot, and computer aided dispatch (911) systems for public safety agencies using a combination of Ruby, PostgreSQL, JavaScript, and Java.

Developed an embedded Linux platform to be hosted at client sites to run portions of this system and provide VPN connections to the primary hosting site.

Responsible for setting up and administering servers for the previously mentioned systems as well as the company's internal servers.

Configured an Asterisk system to handle the company's telephone system, using a combination of analog phone lines along with both hardware and software VoIP phones.

CoreComm Systems Administrator
New Berlin, WI May 1996 – March 2002

Administered legacy ExecPC mail systems with over 100,000 client accounts.

Managed internal and customer mailing lists.

Setup and maintained Unix (Linux, FreeBSD, and Solaris) servers providing a variety of services.

Wrote programs in Perl, Bourne shell, and C to automate administration and provide new services.

Assisted technical support and customer service personnel in fixing customer problems.

Education

Marquette University
Milwaukee, WI — September 1992 – May 1996

Studied Electrical and Computer Engineering, then changed my major to Computer Science.

Activities

I contributed code to implement support for pre-push hooks in git.

I have contributed several enhancements to UltiSnips, a Python-based plugin for Vim; including a complete refactoring of the test system.

I contributed the recording history module for the TivoWeb project, a web-based interface for interacting with TiVo DVRs implemented in TCL.

I have contributed patches to numerous other projects.

Copyright