



Wil (#) Moore III

SENIOR FULL-STACK & DEVOPS & DEVREL ENGINEER



github.com/wilmoore



linkedin.com/in/wilmoore



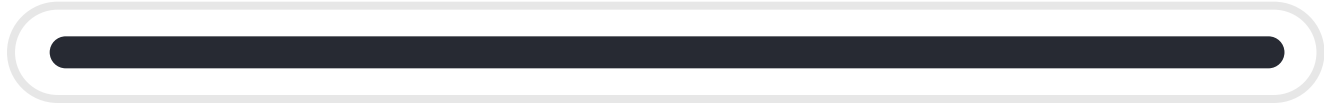
wil.moore@wilmoore.com

Hi, I've been around computers for a long time. I was good with computers long before I was even decent at programming. I'm also an athlete, writer, speaker, mentor and entrepreneur. I've successfully held roles as CTO, Database Administrator, Senior Software Engineer, Full Stack Web Developer, DevOps Engineer, and Developer Relations Engineer.

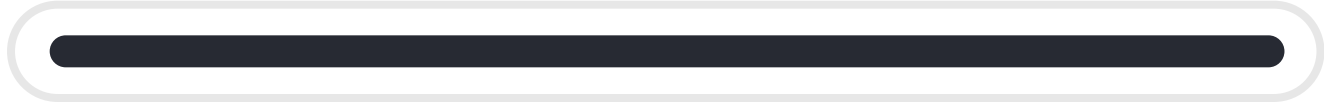
FORMAL EDUCATION

Formal education including high-school, college, and university work. There is no college-level degree as I stopped short to persue entrepreneurial ventures.

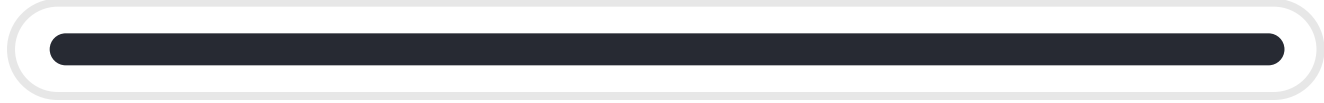
PARAMOUNT HIGH SCHOOL (1992 - 1996 GRADUATED)



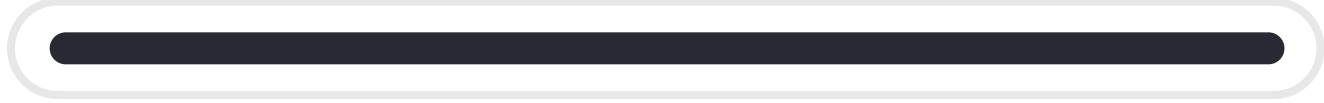
CSULB - COMPUTER SCIENCE & ENGINEERING (1996)



MOUNT SAN ANTONIO COLLEGE (1997 - 1998)



CSULB - INFORMATION SYSTEMS (1998 - 2000)



U. OF PHOENIX ONLINE - INFORMATION SYSTEMS ('02 - '03)



WORK EXPERIENCE

Contract, freelance and direct hire experiences with detailed depictions of **notable projects** I've **delivered** with some client name redactions given a signed NDA.

Sabbatical
Polyglot, LLC

MAR 2022 - PRESENT

Consultanting & Freelance Software Development (SELF-EMPLOYED) AUG 2021 - MAR 2022
Polyglot, LLC

Contract & freelance work is detailed below ... client names are provided where not protected by a signed NDA.

TipOff (tipoffgame.com)
React Native, JavaScript, Fastlane, GitHub Actions

Refactored the TipOff mobile taboo-like party game's codebase to use react hooks which simplified and cleaned up the codebase significantly. The source was written in ES6 JavaScript, React Native, and using hooks. I added fastLane support to enable CI/CD using GitHub actions.

Refactored the entire codebase as several supporting libraries stopped working after moving hooks. This resulted in the code ultimately being a lot simpler than the previous version. The effort ended up being worth it since the game timer was extracted into both a UI and a functional component.

###

EXL
Python, PIP, Poetry, Markdown, Azure DevOps

Created a Python project boilerplate composed of the poetry packaging and dependency management framework, documented its usage with markdown as well as a version for the Azure DevOps Wiki.

Conducted code reviews and high-level architecture review.

###

Client Name Not Disclosed Per NDA (Software Consulting Agency)
React Front-End Page & Component Development, TypeScript & Python Flask API Development, General Debugging

Joined an existing 3rd party software development team that was working on a large SaaS software product. Contributed by autonomously taking on several new front-end page builds with ReactJS and TypeScript. Quickly ramped up by grepping through the codebase with minimal assistance and buitle several new pages and components with markdown documentation and unit tests.

Fixed bugs and made some enhancements in a Flask/Python restful API.

###

Developer Relations Engineer
Electric Coin Company [Zcash] (100% REMOTE)

APR 2021 - AUG 2021

WORKED CLOSELY WITH THE ZCASH/ZEBRA CORE DEVELOPMENT TEAM AS WELL AS THE DEVOPS AND SECURITY TEAMS IN ORDER TO BRING PARITY BETWEEN THE USER DOCUMENTATION, DEVELOPER DOCUMENTATION AND THE ACTUAL STATE OF THE CORE PROTOCOL LIBRARIES AND NETWORK DAEMON. SINCE THE CODE WAS IN A STATE OF TRANSITION BETWEEN THE FORKED BITCOIND CODE (WRITTEN IN THE C PROGRAMMING LANGUAGE) AND THE MORE MODERN RUST PROGAMMING LANGUAGE, I HAD TO GREP THROUGH A LOT OF C CODE IN ORDER TO BRING PARITY TO THE DOCUMENTATION.

I WORKED DIRECTLLY WITH LEADERSHIP TO CURATE NOTABLE EVENTS, MILESTONES, AND ACCOMPLISHMENTS AND SCHEDULE AND COMMUNICATE CONFERENCE CALLS TO THE COMMUNITY.



Wil (₩) Moore III

SENIOR FULL-STACK & DEVOPS & DEVREL ENGINEER



github.com/wilmoore



linkedin.com/in/wilmoore



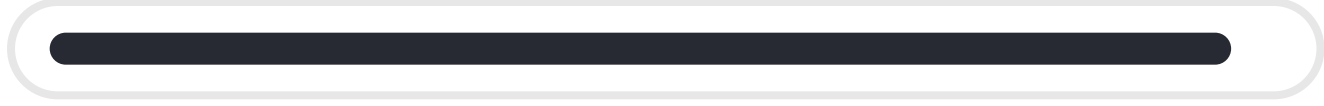
wil.moore@wilmoore.com

Some problems can be solved simply by reading the shell exit codes, thoroughly reading the error message or adding a materialized view to the database.

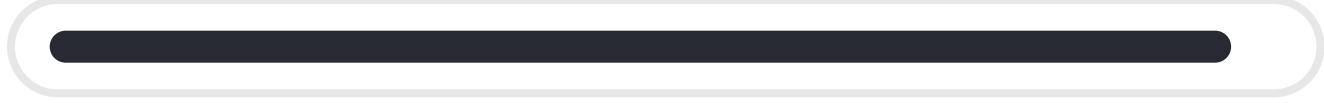
PROGRAMMING LANGUAGES

Languages I've shipped to production, refactored to another language, authored an open-source library or framework for.

JAVASCRIPT (V8/NODE.JS, BROWSER/DOM)



JS COMPILED (TYPESCRIPT, BABEL, COFFEESCRIPT)



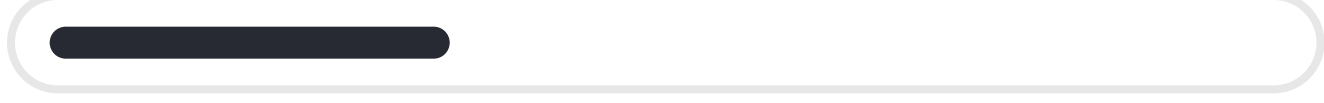
PYTHON



RUST *



ERLANG / ELIXIR



PHP



RUBY



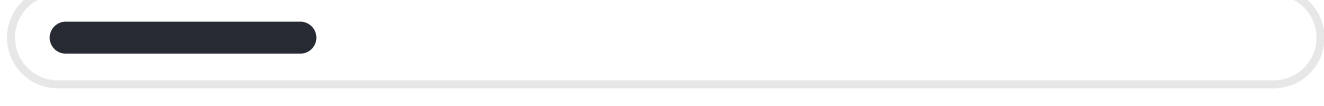
KOTLIN *



GO *



JAVA *



REGEX, AWK, SED



SQL (DDL/DQL/DML, SPROCS, TRIGGERS, VIEWS)



SHELL SCRIPTING (SH, BASH, ZSH, FISH)



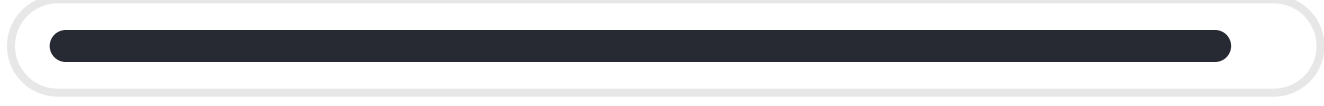
MARKUP / STYLE (HTML/5, CSS/3, SASS, LESS, JSX)



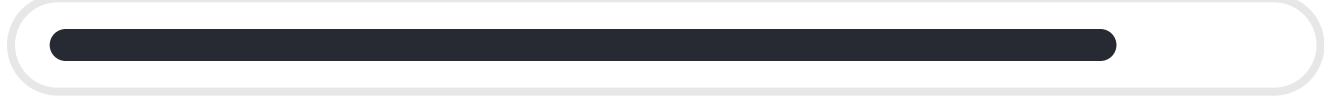
DATABASES

Databases I've deployed and/or managed in production scenarios, or I am actively learning.

RDBMS (POSTGRESQL, MYSQL, SQL SERVER, ORACLE)



NOSQL (DYNAMODB, REDIS, MONGODB)



GRAPH (NEO4J) *



SEARCH (ELASTICSEARCH)



WORK EXPERIENCE

My contract and direct hire experiences with detailed depictions of notable projects I've delivered and in some cases, the reasons behind the tech decisions.

Senior Full-Stack Software Engineer/Devops

(CONTRACT) APR 2019 - MAY 2020

MapQuest (Verizon Location Services) - Denver, CO (40% REMOTE)

I work with data scientists, data engineers, and software engineers to automate and secure instance provisioning, data ingestion pipelines and machine learning workloads on cloud infrastructure.

Setup secure (SFTP) file exchange between data science team and third party data vendors.

CloudFormation, Ansible, Packer, and Jenkins to automate infrastructure provisioning for data science related workloads. #IAC

Replaced data science manual data pipeline with a state machine (AWS Step Functions) based pipeline including integration with AWS Lambda and AWS SSM. #ETL

Kubernetes (K8s) documentation, training, and configuration.

RDS PostgreSQL to Aurora migration. #DATABASE #SRE

Wrote a guide and hosted a primer on Provisioning GitLab in AWS. #TRAINING, #MENTORING, #GITLAB

Senior Full-Stack Software Engineer, devOps, SRE

(CONTRACT) APR 2018 - FEB 2019

Nordstrom - Denver, CO (40% REMOTE)

Designed and implemented a near real-time messaging system with message fanout arriving via SSE. SSE (EventSource) was chosen over Websockets as my experiments and research revealed that SSE is superior in simplicity of debugging, testing, and deployment over Websockets, especially when client-to-API two-way messaging is not needed. Deployment is automated via Terraform and CloudFormation. The RESTful Swagger API (OpenAPI) written in YAML format is fronted by AWS API Gateway and proxies to AWS Lambda which persists messages to DynamoDB for long-term storage, which subsequently emits an event to a DynamoDB Stream and is picked up by an AWS Lambda function which produces to a Kafka topic. The Kafka topic is watched by a Node.js worker process and eventually fanned out to clients via SSE. #APIDESIGN, #BACKEND, #DEVOPS

Designed and implemented an Elasticsearch schema and cluster for a search/filtering strategy which replaced an existing Redis cluster which was previously merging results in-flight which was less than efficient and not very flexible in terms of agility or surviving change. The Elasticsearch cluster was deployed in AWS on EC2 instances. I crafted and documented several Elasticsearch queries per the team's requirements as well as delivered performance metrics via Artillery-based load testing. Integrated a simple Elasticsearch HTTP client along with the queries into an existing Node.js/TypeScript API (not originally maintained by me or my main team – helped another team out with this implementation). #SEARCH

Contributed various production features and enhancements to a Node.js Apollo GraphQL API deployed to AWS ElasticBeanstalk fronted by an Nginx load balancer in a VPC. I increased response throughput by +3x (at least) by way of adding database indices and pagination in hot portions of the source. The performance enhancement was verified with pre and post load testing with Artillery. Wrote custom express middleware as a provider for mocking the calls to the Node.js GraphQL API for Cypress.io testing a React front-end application. #DB, #SCHEMA, #DEVOPS, #BACKEND, #AWS

Authored sections of a Docker/Kubernetes deployment tutorial. Authored an incident response guide: a guide for on-call SREs which included examples of previous incident responses along with screenshots. Built dashboards, monitors and alerts in Datadog and Kibana, with integration to PagerDuty and automated with terraform. #DOCUMENTATION, #DEVOPS, #SRE

Senior Full-Stack Software Engineer

(CONTRACT) OCT 2017 - MAR 2018

Charter Communications - Greenwood Village, CO

Designed and implemented a Node.js API with the IBM/StrongLoop Loopback API framework. API was consumed by a Vue.js single-page-application. Developed custom Loopback middleware and mixins for uploading and persisting attachments in base64 format. #API, #ARCHITECTURE

Using raw SQL DDL, designed and created MySQL database schemas to support new features and enhancements in the API. #SQL, #DDL, #MYSQL

Authored a new developer onboarding guide which covered getting started with contributing to the project, getting around the company's campus, and how to setup the VPN. #DOCUMENTATION



3 of 6

Wil (≠) Moore III

SENIOR FULL-STACK & DEVOPS & DEVREL ENGINEER



github.com/wilmoore



linkedin.com/in/wilmoore



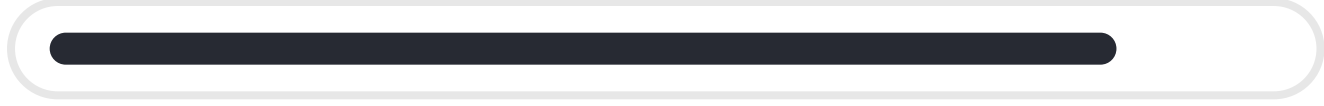
wil.moore@wilmoore.com

I started with desktop publishing, type setting, and graphic design. At one of my first jobs (Kinkos) I debugged issues with copiers, printers, desktop computers, and laptops which gave me an appreciation and some early reps solving problems through troubleshooting.

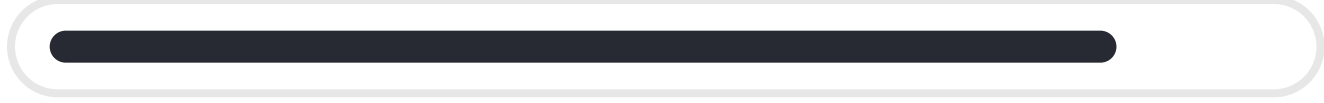
CLOUD TECHNOLOGIES

Technologies that I've worked with in the cloud from the perspective of architecture as well as implementation.

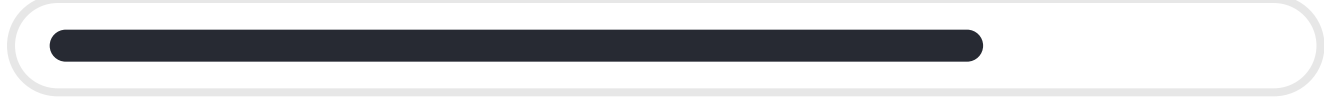
COMPUTE (AWS EC2/ELASTICBEANSTALK, HEROKU)



FUNCTIONS (AWS LAMBDA, SERVERLESS FRAMEWORK)



NOSQL (AWS DYNAMODB, ELASTICACHE)



RDBMS (AWS RDS POSTGRESQL & MYSQL)



STORAGE (AWS S3, RACKSPACE CLOUD FILES)



QUEUE (AWS SQS, RABBITMQ)



MESSAGING (AWS SNS)



WORKFLOW (AWS STEP FUNCTIONS)



IAC (ANSIBLE, CLOUDFORMATION, PACKER)



LOAD BALANCING (AWS ELB)



DNS (AWS ROUTE 53)



CDN (AWS CLOUDFRONT)



MONITORING (AWS CLOUDWATCH, DATADOG, KIBANA)



STREAM (AWS KINESIS, KAFKA)



NETWORK (AWS VPC, SECURITY GROUPS)



WEB SERVICES

Web service (or micro-service) technologies in which I've implemented the producing side (the service itself) and/or a consuming client.

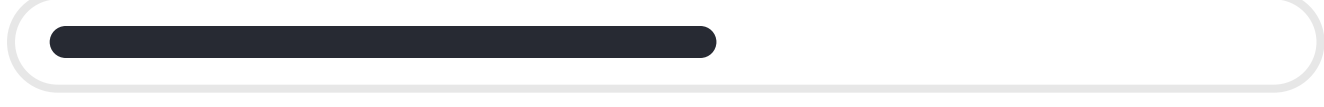
REST & RESOURCE-BASED SERVICES (OPENAPI/SWAGGER)



GRAPHQL (APOLLO-BASED & VANILLA NO-FRAMEWORK)



SOAP (CLIENTS)



WORK EXPERIENCE

My contract and direct hire experiences with detailed depictions of notable projects I've delivered and in some cases, the reasons behind the tech decisions.

Senior Full-Stack Software Engineer
Shutterstock - Denver, CO (80% REMOTE)

(CONTRACT) FEB 2016 - SEP 2017

[API ARCHITECTURE + IMPLEMENTATION, DB SCHEMA DESIGN, DB TUNING] Maintained and refactored an e-commerce **Node.js API** on top of the **express** framework backed by a **MySQL** database using **test-driven development (TDD)**. Implemented a **currency exchange API** with **Ruby** and the **Sinatra** framework backed by a MySQL database using test-driven development (**TDD**).

Contributed to **Node.js orchestration APIs** and built JavaScript **API clients** using test-driven development (**TDD**), including **npm packaging** and artifact release into a **private npm registry**. Evolved the **schema** of a MySQL database through features and enhancements through **DDL** and **DML SQL change scripts**. Introduced a **message queuing** strategy through use of the **RabbitMQ** message broker. Authored design docs, developer guides, and **architecture diagrams** as well as facilitated architecture reviews.

Senior Full-Stack Software Engineer
Aetna (formerly iTriage) - Denver, CO (80% REMOTE)

(CONTRACT) MAY 2015 - FEB 2016

[API, BACKEND, DEVOPS] Implemented and deployed **Swagger APIs with Node.js, Hapi, and Docker**. Contributed to **analytics client libraries** in **JavaScript** and **Objective-C**. Contributed to a **Ruby/Sinatra RESTful API**. Contributed to an analytics platform built on top of **AWS Redshift**. StatsD instrumentation, e2e testing with WebdriverIO/Selenium and **Elasticsearch R&D**.

[DEVELOPER TOOLS] Built reusable project **boilerplates** for **scaffolding** backend **APIs** including documentation with consideration for developer experience (**DX**).

[FRONT-END] Built a secure message center as an **AngularJS** single-page-application (**SPA**) so users could login to view sensitive messages from their health provider in a modern user interface.

Senior Full-Stack Software Engineer
Welltok - Denver, CO

(CONTRACT) AUG 2014 - MAY 2015

[API ARCHITECTURE + IMPLEMENTATION, BACKEND, DEVOPS] Designed and implemented Several micro-service APIs with Node.js and Express which powered the core platform for CafeWell Concierge. Concierge which was comprised of a native mobile application for health care benefits recipients and a web application for administration. APIs were deployed via **Docker** to **AWS ElasticBeanstalk** with some assets being deployed to **AWS S3** and ditributed through **AWS CloudFront**. The administration dashboard was written in **Ruby/Rails** and the **Rails Admin** package with deployment to **AWS ElasticBeanstalk** with **Docker**.

[NLP / AUTO-COMPLETE] Utilized **Elasticsearch's NLP capability** to expose a **Node.js RESTful API** for auto-complete with the **Koa** HTTP library. The **auto-complete** service was consumed by the Concierge mobile application.

[EXTERNAL SERVICE API INTEGRATIONS] As Concierge was meant to be a "cognitive" application, there was some integration with IBM Watson (the cognitive computing engine that that won Jeopardy).

Designed and implemented a **RESTful API** using the **Node.js Koa framework** with integration with the **Active.com API** (http://developer.active.com/docs/read/v2_Activity_API_Search). Packaged as a **Node.js module**, **Open-sourced the code**, and published to the **npm registry** (<https://www.npmjs.com/package/active-activity-search>).

[ETL] Migrated a legacy Python **ETL** project to **data transformations** powered by **Node.js** streams. **Packaged the core helpers as a Node.js module**, **Open-sourced the code**, and **published to the npm registry** (<https://www.npmjs.com/package/text-pipe>).

Senior Front-End Software Engineer
Time Warner Cable - Broomfield, CO

(CONTRACT) AUG 2012 - JUL 2014

Built an **HTML5 single-page-application (SPA)** and **HLS streaming video player** with custom scrubbing features. This system was targeting a **whole-home IP TV** (cable set-top box) system targeting millions of customer homes. Utilized **test-driven-development (TDD)** to implement and deliver features, bug-fixes, enhancements, and performance tuning.

Automated testing w/ **Mocha, Jasmine, Sinon, Chai**, and **PhantomJS**. Build automation with **GNUMake** and **CI/CD** with **Jenkins**. **IaC** via **Puppet**. **Node.js** for utility scripting, testing, and backend service stubbing.

Designed & Developed various HTTP API clients using **jQuery AJAX, Superagent, Promises**, higher-order functions, and "README"-driven development. API clients for retrieving and playing HLS streams as well as clients for retrieving emergency (**EAS**) notifications and application updates. Originally utilized **WebSockets** which were switched to **Server Sent Events, Mocha, Grunt, Gulp**, and **GNUMake**.



4 of 6

Wil (#) Moore III

SENIOR FULL-STACK & DEVOPS & DEVREL ENGINEER



github.com/wilmoore



linkedin.com/in/wilmoore



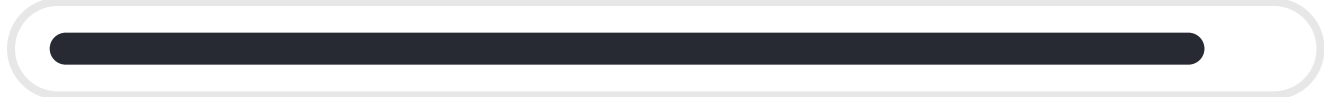
wil.moore@wilmoore.com

Redirection is one of the most powerful, yet underappreciated computing concepts ever invented.

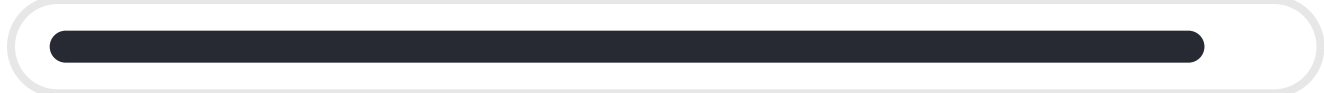
TEST FRAMEWORKS

Automated test frameworks that I have used and feel comfortable writing new or modifying existing test cases.

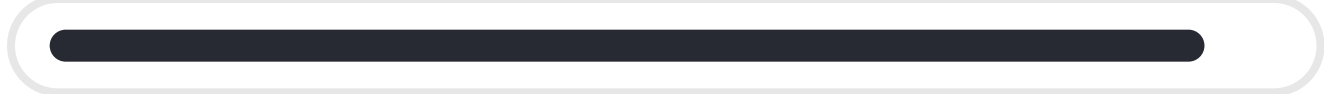
JS (TAPE, MOCHA, JASMINE, TESTEM, KARMA, JEST)



RUBY (MINITEST, RSPEC)



PHP (PHPUNIT)



API FRAMEWORKS

API libraries and frameworks that I've worked with.

NODE.JS (EXPRESS, KOA, FASTIFY, CONNECT, HAPI)



NODE.JS (APOLLO GRAPHQL, STRONGLOOP LOOPBACK)



NODE.JS (NODE STANDARD LIBRARY HTTP/HTTPS)



RUBY (SINATRA)



WEB & MOBILE FRAMEWORKS

API libraries and frameworks that I've worked with.

REACT, REACT NATIVE



VUEJS



ANGULAR



BACKBONE



CI/CD

Continuous Integration and continuous deployment systems that I've used, setup, and/or maintained.

JENKINS



GITLAB CI



SEMAPHORE, CIRCLE, TRAVIS



WORK EXPERIENCE

My contract and direct hire experiences with detailed depictions of notable projects I've delivered and in some cases, the reasons behind the tech decisions.

Senior Full-Stack Software Engineer

* DEC 2011 - JUL 2012

Net-Results - Golden, CO

[FRONT-END, BACK-END, UI/UIX, DEVOPS] UI/UX design and implementation for a multi-tenant browser-based **single-page-application (SPA)** for a customer **cloud file management** interface using **AWS S3** for file storage with a **PHP 5.3** backend.

[FRONT-END, BACK-END] Contributed to a **REST API** for **account provisioning** and client maintenance with **jQuery** and backed by **MongoDB**. Contributed to a **multi-tenant** browser-based **single-page-application (SPA)** for building marketing landing pages using **drag-and-drop** using HTML5, CSS3, and **YUI3**.

(Team Lead) Senior Full-Stack Software Engineer

JUN 2010 - AUG 2011

Navigant (formerly Ethos Partners) - Denver, CO

[API ARCHITECTURE + IMPLEMENTATION, BACKEND, DEVOPS] Lead a team of 3 developers in successfully re-architecting a legacy monolithic web application that **allowed health care providers to create health risk assessments (HRA)** with custom business rules, **branching**, and demograph intake.

The original software was written with PHP 4.x which my team replaced with **PHP 5.x** using the **Zend Framework** and increased test coverage to **95%** (an increase of **94%**).

Full-Stack Software Engineer

(CONTRACT) OCT 2009 - JUN 2010

Kaango - Greenwood Village, CO

Designed, implemented, and documented a Full-Stack application utilizing **PHP**, **HTML**, **JavaScript**, and **CSS** for **managing classifieds**. Implemented a job queue on top of **AWS SQS (Simple Queue Service)** for handling batches of classified ads which were **transformed** before being persisted into a **MySQL database** as well as indexed into **Solr** for **full-text search**.

Designed, implemented, and documented a Full-Stack application utilizing **PHP**, **HTML**, **JavaScript**, and **CSS** for a **question and answer website** (similar to StackOverflow). Implemented an authentication and **RBAC** (Role Based Access Control) system in **PHP**. Refactored poorly performing **SQL** queries by **tuning indexes**. **Increased unit test coverage** from 3% to approximately 80%.

Full-Stack Software Engineer

(CONTRACT) MAR 2009 - OCT 2009

Reed Business Information - Highlands Ranch, CO

Introduced unit testing to the team. Re-factored tightly coupled code with **inversion of control** for better testability. Successfully recommended and utilized **cross-browser testing** tools. Implemented an **Object-Oriented JavaScript** based tool, which maps a list of valid **CSS definitions** to be **imported** and **exported** in bulk. Successfully **streamlined** the development team's **hiring processes**. Lead and **organized weekly user acceptance** testing **walkthroughs** for new features and enhancements.

Contributed to **launch** of a number of **high-traffic websites** such as: **furnituretoday.com, designnews.com, multichannel.com, controleng.com, mbtmag.com, mbtmag.com, and controleng.com**.

Database Administrator, ERP & Subscriptions MGR.

NOV 2006 - MAR 2009

Gaiam, Inc - Louisville, CO

Managed continuity/subscription systems for **spiritualcinemacircle.com**. Developed a **SOAP** interface for retrieving data out of the **ECOMETRY ERP** system. Migrated an **ECOMETRY** system from **HP 3000 MPE/iX** to **Windows/MSSQL 2005**. Ported **Suprtool ETL** scripts to **SQL**, **DOS batch**, **AWK**, and **Python** scripts. Implemented **configuration change management** on **Subversion**. Integration with remote web services (**SOAP** and **REST**) including **email subscription** list synchronization. Managed the integration of **ECOMETRY** with a 3rd party fulfillment provider through **batch processes** and **webhooks**. This included a web-based **AJAX** interface for the fulfillment staff to **review** and **confirm** shipments.



5 of 6

Wil (#) Moore III

SENIOR FULL-STACK & DEVOPS & DEVREL ENGINEER



github.com/wilmoore



linkedin.com/in/wilmoore



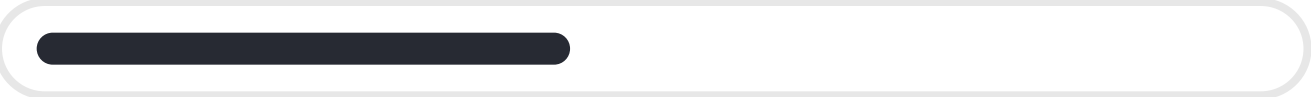
wil.moore@wilmoore.com

Did you know that I've published several open-source GitHub repositories. JavaScript, TypeScript, Bash, PHP, Erlang, GNU Make, and some technologies I've completely forgotten about. I have over 70+ modules on npm. Check those out here <https://www.npmjs.com/~wilmoore>. Lots of them are written in a functional data-last style. Many of them published before Lodash and React made fp in JS cool.

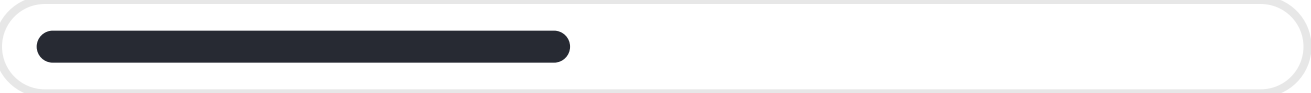
WEB MAPPING

Web mapping libraries, frameworks, and Web APIs that I have worked with.

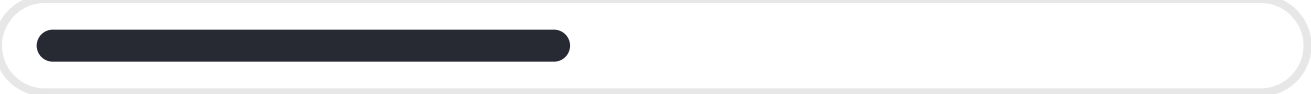
MAPBOX



LEAFLET



WEB APIS (GOOGLE MAPS, MAPQUEST)



CONTAINER PLATFORMS

Containerization platforms, including container orchestration platforms that I've worked with.

DOCKER



KUBERNETES (DEV DEPLOYMENTS)



MESOSPHERE (DEV DEPLOYMENTS)



GRAPHICS, UI, DIAGRAMS

Vector graphics, photo editing, illustration, desktop publishing, UI prototyping, and diagramming programs I've worked with.

VECTOR (SKETCH, ILLUSTRATOR, AFFINITY, FREEHAND)



PHOTO (PHOTOSHOP, PIXELMATOR)



DIAGRAMMING (OMNIGRAFFLE, BALSAMIQ, LUCIDCHART)



PUBLISHING (PAGEMAKER, INDESIGN, QUARKXPRESS)



NOW LEARNING *

Technologies that I am currently actively learning are listed here. Because I'll run out of space here, look for items with an asterisk * next to them. Those are items that should also be here.

SOFTWARE ENGINEERING MANAGEMENT *



DEEP LEARNING *



DATA SCIENCE *



KUBERNETES OPS WITHOUT HIGH-LEVEL TOOLING *



MY PRODUCTS

Entrepreneurial product ventures as well as notable things I've published...

CTO, Senior Full-Stack Software Engineer TipOff, LLC.

JAN 2019 - OCT 2020
tipoffgame.com

As the temporary **CTO** of a **mobile game** company called **TipOff** I re-architected and re-skinned the iOS and Android apps and re-published the apps to the app stores using the **React Native platform**.

I originally utilized expo; however, that proved to be less than maintainable (I wrote an article detailing issues with expo on LinkedIn). I eventually replaced **expo** with **fastlane**. Unfortunately, this change meant the entire app needed to be re-written from scratch using the non-expo path. New libraries needed to be selected, installed, and tested.

I was able to re-build the game from scratch in about 2-weeks. I had a working game ready to be published for **iOS**; though, I concede that my builds were rejected several times for over one week before being accepted. I ran into various issues, mostly to do with screenshots; however, I also once attempted to publish an erroneous build. Previous **exposure to React** and the **npm ecosystem**



Wil (≠) Moore III

SENIOR FULL-STACK & DEVOPS & DEVREL ENGINEER



github.com/wilmoore



linkedin.com/in/wilmoore



wil.moore@wilmoore.com

Unicode & UTF-8 is one of the most elegant "hacks" invented in computers that we all mostly take for granted.

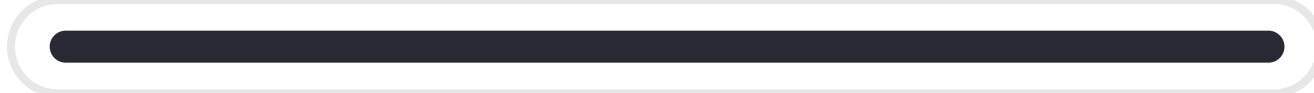
CERTIFICATIONS

Technical certification exams that I've sat for and passed. A * is used to denote technical certification exams that I am actively studying to sit for this calendar year.

ZEND CERTIFIED ENGINEER - 2010 (ID: ZEND015278)



MICROSOFT CERTIFIED PROFESSIONAL (ID: 2592587)



AWS CERTIFIED SOLUTIONS ARCHITECT ASSOCIATE *



OPEN SOURCE, ARTICLES, CONFERENCE TALKS

Open-Source projects I've authored, articles I've written on software development, and conference talks I've given that may be moderately interesting to mention.

selectn.js

npmjs.com/package/selectn

RELEASED: MAR 2013

~368K downloads per month

selectn.js is a curried **property accessor** function written in **JavaScript** for **Node.js** and the **browser** that resolves **deeply-nested object properties** via dot/bracket-notation string path while mitigating **TypeErrors** via friendly and composable API.

I wrote **selectn** because I noticed that **most of my colleagues** on the **HNAV** project during my time at **Time Warner Cable** were **repeating the following pattern** throughout the codebase:

person && person.info && person.info.name && person.info.name.full

selectn.js allows you to refactor the above **to the following**:

selectn('info.name.full', person)

php-version

github.com/wilmoore/php-version

RELEASED: JAN 2012

@ GitHub 582 stars, 43 forks

php-version is a command-line **PHP version manager**. I wrote **php-version** so as to easily switch between different versions of PHP since at the time I was frequently switching between different client PHP projects. Though I've personally moved past using version managers as part of my everyday development workflow, I still maintain it as it's useful to many developers. I plan to re-factor it away from being a bash shell program to a compiled binary written in **rust**.

uuid-regex

npmjs.com/package/uuid-regex

RELEASED: JAN 2016

~5K downloads per month

uuid-regex is a **JavaScript** regular expression (**RegExp**) for finding an **RFC4122** compliant **UUID** in a string. While it may seem unintuitive that something as tiny as a single regular expression would need to be open-sourced, consider the fact that this problem no longer needs to be solved in the JavaScript ecosystem and if I missed some edge-case, it will be resolved in a single place.

curry2

npmjs.com/package/curry2

RELEASED: MAY 2015

~100K downloads per month

curry2 curries a binary function, also said to be a function that expects 2 parameters or arguments. **curry2** is useful in **JavaScript** because unlike **true functional languages** where functions are **automatically curried** (i.e. **Scala**, **Haskell**), in order to approximate the same, a lot of boilerplate code must be written to wrap the function you want curried. There are other libraries that will wrap a function so it acts like a curried function; however, most of them trade-off clarity and performance for convenience of currying any number of parameters. **curry2** makes the trade-off of only supporting functions with an arity of 2 (thus a **binary function**) as I realized that I never had the urge to curry anything other than a **binary function**. You might also like to check out the following article that I wrote back in 2016 about some of the benefits of currying in JavaScript.

https://medium.com/@wilmoore/un-bind-your-js-with-curry-a8657a4138cb

Confoo Montreal Conference

https://confoo.ca/en/speaker/wil-moore

2012 & 2013 Conferences

https://joind.in/user/wilmoore

Conference Talks

10 advanced, yet digestible SSH techniques
speakerdeck.com/wilmoore/10-advanced-yet-digestible-ssh-techniques

Painless, version-controlled database refactoring
speakerdeck.com/wilmoore/painless-version-controlled-database-refactoring

DIY Private Cloud w/ VirtualBox and Chef
confoo.ca/en/2012/session/diy-private-cloud-w-virtualbox-and-chef