

Steven Wolfe

Email: steven@swolfe.me - Website: <https://swolfe.me> - Github: <https://github.com/s32x>

Objective

To obtain a Software Engineering position with a fast paced, leading edge company where I am able to contribute to the growth and success of a top tier engineering team by leveraging the latest in software development technologies.

About Me

I'm an extremely passionate software engineer who loves building large scale software applications. For the past four years I've been working as a backend Software (primarily Go) Engineer for both large and small startups and have experience on a variety of different cloud platforms. Writing clean, documented, and efficient Docker based services is something that I take a lot of pride in and often spend a fair amount of time outside of the office perfecting my craft. Below is my Professional and Educational backgrounds, as well as a collection of personal projects that better demonstrate my experience.

Professional Experience (4 Years)

Tile38 - <https://tile38.com>

Software Engineer

August 2018 - April 2019

- Worked toward building a commercialized cloud offering of the Tile38 Geospatial database.
- Implemented metrics systems such as Prometheus to facilitate database monitoring solutions.
- Authored an interactive demonstration of using Mapbox and Tile38 to efficiently plot 11M points - <https://geonames.tile38.com>
- Contributed to the open-source Tile38 database repository on Github

Brightcove - <https://brightcove.com>

Software Engineer

September 2016 - August 2018

- Worked in a scrum environment to write well tested Golang microservices on the Server Side Ad Insertion team
- Contributed to the completion and launch of Server Side Ad Insertion on the BOLT platform
- Helped integrate tooling that was instrumental in the migration of legacy customer video data
- Converted several services from Negroni to Go-Kit

Control Concepts - <http://controlconcepts.net>

Junior Developer

April 2016 - September 2016

- Collaborated with the core software development team to complete NYC based client software projects
- Utilized languages such as C# to write Crestron Fusion client libraries
- Experience with ASP.NET, and Razor for full stack client applications

Teltech Systems - <https://teltech.co>

Software Engineer Intern

May 2015 - February 2016

- Wrote countless backend Docker microservices in both Golang and Python
- Frontend AngularJS experience on multiple telephony based applications
- Worked primarily in a test driven environment to ensure a robust backend
- Key contributor to both frontend and backend of apps such as TrapCall, RoboKiller and WhoApp

Education

William Paterson University (2.93 GPA)

Bachelors of Science in Computer Science, Fall 2015

- Earned Dean's List in Fall 2012

Projects

Trumail - <https://trumail.io>

Open-Source Email Verification API

- Written in pure Go as a Docker microservice that is built and pushed to Dockerhub using CircleCI
- Usage of the SMTP protocol to send RCPT commands to MX servers
- Ranked #5 Product of the Day on ProductHunt on October 16th, 2017
- Reached top position in Google for the keyword "email verification api"
- Domain and Github repository sold to one of the longest standing email verification companies in the world - EmailChecker (EmailVerify LTD)

Clipr - <https://clipr.tv>

Twitch Clip Aggregation Webapp

- Written in pure Go using nested go-templates
- Hooks into the Twitch JSON API to retrieve top clips, games and channels
- Converted from a Docker based service to a serverless function running in AWS lambda using the Up deployment tool
- In memory and Cloudflare CDN caching help to reduce load on the origin functions

Anirip - <https://s32x.com/anirip>

Open-Source Crunchyroll episode/subtitle ripper CLI

- Utilizes the goquery web-scraping package to retrieve show/episode data
- Uses Go's http clients cookiejar to store authentication cookies and retrieve top quality video streams
- Programatically calls the ffmpeg CLI to dump scraped HLS stream URLs to disk

Ovrstat - <https://ovrstat.com>

Open-Source Unofficial Overwatch Stats API

- Utilizes the goquery web-scraping package to translate the Overwatch player stats page into a structured output
- Returns a large JSON response payload and supports both Console and PC players

IPData - <https://ipdata.info>

Open-Source IP Geolocation API

- Utilizes the free MaxMindDB City and ASN datasets to query and return in JSON format Geolocation and ISP data given any V4 or V6 IP address

Gamedetect - <https://s32x.com/gamedetect>

Open-Source Game Detection API

- Utilizes a trained neural network to classify game screenshots that are within the top 100 popular games on Twitch
- Uses a python script that uses OpenCV to extract frames from random streams on Twitch and then builds a trained graph
- The trained graph is served via HTTP using the Go Tensorflow package which responds to POST requests that contain a relatively clear game screenshot

Extra

References and phone number are available upon request