**Charles Fenwick Elliott**

[Charles@fenwickelliott.io](mailto:Charles@fenwickelliott.io)

Fenwickelliott.io

(917) 365-9424

Github.com/FenwickElliott

LInkedin.com/in/FenwickElliott

**Education**

* Bsc Computational Chemistry, University of Bristol, 2007 – 2010
* Bloc.io Code school, 2017

**Projects**

**ElectroBullet**

HTML5/CSS, JavaScript, Electron

A SMS desktop application utilizing the PushBullet service.

Built the entire project from conception to completion independently, with no non-native dependencies and ‘use strict’. It required some reverse engineering (and traffic snooping) to fill in the gaps of their deprecated API documentation and some lateral thinking to work around issues like the PushBullet authorization page being incompatible with Electron.

The project required, OAuth2 user authentication, setting up a local host and parsing data received through it, persisting data in application storage, creating custom promise chains with resourceful error handling and optional object literal parameters, websocket management, binary encoding, custom menu creation, recursive user data destruction and cross platform packaging, distribution and native notifications.

**Snatch**

Ruby, Rails, Rack, Go

Implementing the same simple task in an A/B comparison context.

Snatch is a simple application with a simple goal; to add the Spotify’s currently-playing track to a designated playlist. The idea is that I am usually listening to music while performing other tasks and I wanted the ability to make a record of something I liked without breaking concentration for more than a second. So once initialized the application has to run to completion with a single command.

Since I first constructed Snatch (for iTunes) with applescripts when I was a child it has served as a personal ‘Hello World’ when I am familiarizing myself with a new environment because it covers some significant ubiquitous components such as, REST calls and API consumption, user authentication (and OAuth2), data persistence and cross platform distribution.

RailSnatch

Rails provided great insight into the ease and shortcomings of working within a framework, the convenience of convention over configuration and the wealth of relevant gems makes for a rapid development environment but I disliked the limitations, inefficiencies and lack of involvement in the underlying methods.

In addition to consuming the Spotify API I wrote and hosted my own API in order that users could download a generated script and trigger a snatch with a single rest call, rather than visiting the website or refreshing the API token client side.

RackSnatch

Writing my own framework and ORM gems from the ground up, re-implementing Snatch and serving it globally on a Raspberry Pi provided a much more comprehensive understanding of the structure, workings and practicalities of hosting of web apps.

GoSnatch

Working with data at the byte and buffer level proved invaluable in fostering (and cementing) a true understanding of lower level methodology. While the unforgiving nature of Go provided some challenges the level of control, efficiency and elegance of Go proved very rewarding. After working in Go I used the lessons learned to refactor several projects and it certainly lead me to look at new projects very differently.

GoSnatch required handling concurrent routines and channels, varying local storage location by environment, marshalling interfaces of varying and unpredictable data types, cross platform compilation and distribution, OAuth2 authentication with a temporary local server and handling native notifications all without any non-native module dependencies.

**DevOps**

In order to attain a full understanding of both extremes of DevOps I wrote and loaded a static website onto a raspberry pi and served it globally with a custom domain through my home IP address with a homemade protocol for notifying myself incase my (non-static) IP address ever changed.

Then I wrote a responsive site and launched it through AWS with HTTP/2, TLS 1.2 incorporating S3, SES, Route 53, CloudFront, WorkMail, IAM authentication, invalidations, analytics, versioning and server side encryption.

While DevOps is not something I am particularly passionate about understanding all facets of whatever I am working with is.

**Other Experience**

**Just-Tech**

A small tech service provider where I worked on all facets of researching, implementing, debugging and maintaining tech services for legal non-profits, not least NYLAG.

**Hospitality**

Several years working in bars and restaurants, starting as a bar-back and working up to general manager. I have many excellent references testifying to my non-technical abilities and character traits.

**Technical Skills**

HTML, CSS, JavaScript, Ruby, Rails, SQL, Go, DevOps, AWS, Electron, OAuth, REST API Design and Consumption, Git, Bash.