Tobias S. Davis

(402) 468-8404
tobias@davistobias.com
8103 Templeton Drive
Omaha, NE 68134

Guiding Principles:

- I want to always be learning new things.
- I value finding the right solution that fits the need.
- I want to make things that users care about.
- I want the things I make to be high quality.
- I want to work with other people who care about those same things.

Professional Experience:

From Now On (Omaha, NE): September 2016-Current

Our clients are schools and event organizers. We create a custom mobile app for their fans, providing live event information such as gameplay information, audio streams, and more.

As a full-stack developer, I build and maintain the web UI of the content management system that the clients interface with, as well as the architecture, design, and maintenance of servers/services used in our system. These services are primarily written in JavaScript, and run in the Azure and AWS environments.

When I arrived at From Now One, I inherited a large and incomplete code refactor and re-architecturing project, which I took from an unusable state to a fully functioning, mostly tested webapp now used in production.

The web application and services were largely untested, but I introduced automated tests on all pull requests with code linting as part of the tests. This has reduced our bugs and regressions significantly.

Since I started, I have been the lead webapp engineer and am currently the lead JavaScript engineer. As part of my comittment to quality, I introduced a functional composition style of coding, which has made our code many times easier to understand and maintain.

Currently my work load is balanced between building new features and migrating old code and old architecture to best-practices design, which eliminate bugs and technical debt.

My latest project is a large re-architecture of our backend, taking our existing Azure-based services and migrating them to AWS, while also making the API interfaces more consistent. This has been a difficult but rewarding experience, so far!

eDataSource (New York, NY 10001): May 2012-September 2016

The overview is that I helped develop software for analysis of email data, using aggregated data gathered from a panel of over 1 million active users.

In my first couple years I primarily designed tools which analyzed emails, parsing financial data from receipts. The primary difficulty was that, since the emails were from third-party senders, the content was not in my control and was often broken and faulty. Despite this, the accuracy of the tools I designed was high enough that the tools themselves and the aggregated data was still being used as a predictive tool for financial investors several years later.

The generated aggregate data was stored in MySQL, in tables of many millions of rows. As part of the QA process, and as part of client driven requests, I was responsible for writing queries which were performant given very large tables and inadequate indexes.

In late 2013 I was given an opportunity to switch my focus to the front-end webapp that eDataSource had been developing. The website primarily used AngularJS and Highcharts to present email information to our clients, helping them determine effectiveness of email campaigns.

Early on I was the lead on two big and painful software upgrades (Bootstrap 2.2 to 3.3, AngularJS 1.1 to 1.3) and I'm pretty proud of how smoothly that went. The upgrade process spanned at least one full month and involved refactoring and changing hundreds of HTML and JS files, and tracking down dozens of difficult to find bugs.

I was also the front-end lead in a two-person team for the design of a new part of the webapp, which helped clients test email campaigns prior to sending them to users. This design had about 8 new screens, several new complex UI objects, and required the design of several new API endpoints, and I'm excited about how smoothly that project went.

In early/mid 2016 I worked with a team of 2-3 other developers creating a redesign of one of our webapps, where we built a set of static pages and one single-page webapp, accessed by thousands of users per day. It was very satisfying to see our signup numbers increase and churn rate drop drastically after the release of the new product.

As part of the Boxbe webapp redesign, I implemented and wrote the code for the client and server side of the OAuth flow for Google, Yahoo, and Microsoft/Outlook, along with a careful test suite for it.

Toward the end I worked with a team of 2-4 in designing a new screen for our analytics webapp, which took aggregated data about a company and presented a view comparing that company's metrics with the metrics of other companies, to let our clients know how their email campaigns were performing relative to their competitors. The feedback from our clients was very positive.

Although I was not the sysadmin, since our team was only 3-5 people until the end of my time at eDatasource, I had to learn the sysadmin side of things as well. In particular, I was familiar with all the big AWS offerings. We used SQS to pass around many millions of messages per month, Aurora to hold terabyte sized databases, auto-scaled EC2 to run our different webapps and various other tools, and S3 and Glacier to store hundreds of millions of files.

While working at eDataSource, I built things using:

- Client Related: AngularJS, Riot.js, Highcharts, Browserify, Bootstrap, Babel (ES6 to ES5 mostly), PreCSS, PostCSS, LESS, JS, HTML, CSS, XML, Socket.io, Moment.js, jQuery, ACE (web code editor).
- Server Related: Java 1.6-1.8, Amazon Web Services (AWS, including EC2, S3, SQS, SNS), Google Guava, GSON/JSON, MySQL, HyperSQL, Redis, MongoDB (only in passing) JDBC, JAX-RS, Jetty, Jersey, Maven, Joda (date library), Log4J/SLF4J, OAuth, YAML, JMock, Spring Framework, Atmosphere (Java websocket implementation), Swagger.io, PhantomJS.
- Development Tools: IntelliJ, Eclipse, JRebel, bash, Node.js, npm, LiveReload, browserify, ES6 and ES5, minify/uglify for JS, Mustache (template language), Tape (test framework), BrowserStack, Markdown to HTML transforms, Git, Github, Bash, Ruby, Homebrew.

Prior to eDataSource:

My professional career began at eDataSource, but prior to that I was doing some private contract work with WordPress and PHP.

For about a year, during college, I built custom WordPress templates and plugins for a client, and for some friends pro bono. I still maintain and update several of those sites.

Personal Projects:

Although I do learn quite a bit while working, my personal philosophy is to always be learning new things. Outside of work hours I build modules and websites, exploring new technology primarily in the JavaScript/npm world.

I've made and maintain a number of npm modules. A few that I am pretty happy with are:

- **imap-tools:** Small collection of tools making it easier to work with IMAP in JavaScript.
- **hex2words:** Given a hexadecimal value, it returns a list of PGP Words. This is a list of words two users can read to verify they have identical signatures, for example.
- **json-schema-to-markdown:** Turn a JSON schema (a JSON object used to validate other JSON objects) into a human-readable markdown file. It makes writing documentation a lot easier.
- **pipe-transform-cli:** A command line utility I wrote that lets you stream in one encoding and stream out another encoding. For example, hex in and base64 out.
- **ftp-core:** Although the project has been set aside, this was an attempt to create a pure JavaScript implementation of the FTP protocol. It was really fun to take the RFC specs and turn them into code. Sadly I ran out of time to finish it.

Tech Stack for Personal Projects:

I've created tools and products using many of these buzzwords:

- SvelteJS, AngularJS, Ractive.js, Highcharts, RollupJS, Browserify, Bootstrap, JS (modern and historic), LESS, HTML, CSS, XML, Socket.io, Moment.js.
- Server Related: Express, Ecstatic, MySQL, OAuth, YAML.
- Development Tools: Sublime Text, bash, Node.js, npm, browserify, JS cli tools (minimist is nice), minify/uglify for JS, Mustache (template language), Tape (test framework), Markdown to HTML transforms, Git, Github, Bash, Ruby, Homebrew.
- Operating Systems: Windows, Mac, Ubuntu, Debian.
- Other Things: SQLite, PHP, I spend a lot of time learning basic cryptography techniques (AES, RSA, SHA, SHA-2, PGP/GPG), Swagger.io, JSON Schema, I've contributed to ngbp (AngularJS boilerplate), Noddity, and have contributed to PhantomJS.