

# Dylan Compton

## Software Engineer

[dylan.e.briar@gmail.com](mailto:dylan.e.briar@gmail.com)  
[github.com/dylanbriar](https://github.com/dylanbriar)  
[linkedin.com/in/dylanbriar](https://linkedin.com/in/dylanbriar)  
[dylanbriar.com](https://dylanbriar.com)  
Austin, TX | 512-757-2682

## Technical Skills

---

- Languages and Frameworks: *TypeScript, JavaScript (ES6), GraphQL, React, Express, Jest, Vitest, SCSS, CSS, HTML5*
- Technologies: *Redis, Redux, SQL (PostgreSQL), NoSQL (MongoDB), Node.JS, Bun, Vite, Webpack, Docker, CI/CD (Netlify)*

## Experience

---

### Software Engineer | *BuQL (Hosted by OSLabs)*

2024-present

- Built the product in **Bun** to introduce a server-side GraphQL caching solution and, through Bun's use of the JavaScriptCore engine and Zig language, optimized start and response times from the database or cache.
- Created the middleware package using **GraphQL** to establish predictable API management, leveraging GraphQL's avoidance of under- and over-fetching to support partial querying, such as 67% from the cache and 33% from the database.
- Cached responses via the **Redis** ioredis to achieve ~1000% faster response times and, through Redis' agnostic approach to data structures, be considerate of the data types that BuQL users may send.
- Integrated **Vite** for better build times and HMR speeds, prioritizing performance in the npm package; Vite also introduced Rollup's tree-shaking algorithm for asset optimization (~8% smaller bundle sizes for medium-sized applications).
- Stored data in **MongoDB** with Mongoose as the ORM for the front end, allowing for schema flexibility and potential horizontal scaling for more advanced demonstrations; MongoDB's focus on DX sustained productivity during development.
- Wrote tests via **bun test**, enabling custom match extensions for future iteration and independence from cumbersome testing frameworks such as Jest (40% size increase), without sacrificing versatility, such as UI and DOM testing.

## Open Source

---

### Software Engineer | *Cyber Stars*

2024

- Implemented **PostgreSQL** to store 33% of the structured and predictable data retrieved from the NASA API, taking into consideration the read-heavy nature of the app and the data's predictable types and schema structures.
- Tested authentication with **Vitest**, for competitive testing speeds, as well as the simplistic configuration that better the DX; this permitted a multithreaded approach while avoiding an encumbered application (34% smaller than Jest equivalent).

### Software Engineer | *coolercards*

2024

- Iterated on a middleware design pattern established through **Express** for HTTP/API endpoint requests, allowing for code modularity via router and controller files that, alongside the Express informational ecosystem, kept debugging more straightforward.
- Improved test code coverage with **Jest** by 25%, testing the back-end before continuing on the iteration; the minimal configuration and the shortened feedback loops from parallel execution enabled an expedited iteration process.

### Software Engineer | *The Stockyards*

2024

- Implemented **React** to make 5 components, such as the user's portfolio and news display, and take advantage of the virtual DOM to provide efficient, browser-agnostic DOM manipulation, aligning the application with Flux architecture.
- Deployed **Webpack** to maintain flexibility for plug-in options and configuration in anticipation for iteration; the accommodation to ES6+ syntax made it particularly suitable as the Node.js application's HMR and code uglification tool.

## Education

---

- Honors Student, Honors Ambassador | *Austin Community College*

2022-2023

## Publications / Talks

---

- Presenter, Author | [Docker Image Security](https://www.youtube.com/watch?v=I5o_LcOS674) | [https://www.youtube.com/watch?v=I5o\\_LcOS674](https://www.youtube.com/watch?v=I5o_LcOS674) 2024
- Presenter, Co-Author | [CI/CD Security](https://www.youtube.com/watch?v=ybMFv9gRJoU) | <https://www.youtube.com/watch?v=ybMFv9gRJoU> 2024
- Author | [Looking for a GraphQL caching solution in Bun? It's time to BuQL Up](https://tinyurl.com/BuQLArticle) | <https://tinyurl.com/BuQLArticle> 2024

## About Me

---

Free soloing in Austin; tastefully smack talking fantasy football league competitors (NFL); joining fantasy roleplaying campaigns that get rescheduled to oblivion; jotting down ideas that gather dust in a notes application