# David Chai Software Engineer

347.276.8760 | davidchai717@gmail.com | linkedin.com/in/davidchai717 | github.com/davidchai717 | New York, NY

**Strong**: JavaScript (ES6+), React (Hooks API, Context API), Redux, Node, Express, SQL (Postgres), NoSQL (MongoDB & Mongoose), Authentication (Bcrypt, Express Session), Version Control (Git, Github), RESTful APIs, Agile/Scrum Development
**Experienced**: Webpack, Docker, Travis CI, TDD (Jest, SuperTest), AST (Acorn), D3, React Router, WebSockets, Microservices, gRPC, AWS

## Experience

**Reactime** | Software Engineer | *Featured on Top 5 React Newsletter* 2019 - Present

- Developed a React state management dev tool, comprised of an NPM package and a browser extension, that automatically records each state change from users and allows developers to switch between various state changes to enhance the debugging process
- Improved Reactime's React tree traversal algorithm by following React's underlying Fiber structure in order to identify functional components using the useState or useReducer Hooks and then subscribe to the user's usage of the built-in dispatch function using an event emitter, allowing the NPM package to capture and store state changes automatically for components employing React Hooks
- Implemented React Hooks snapshot reverting support by creating an algorithm that would recursively apply selected state snapshot data from Reactime's state cache to user's application, allowing for real-time browser rendering of the desired state snapshot
- Leveraged Acorn's native JSX support to parse user's React components utilizing Hooks into Abstract Syntax Trees (AST) in order to dynamically retrieve user-declared variable names of Hooks and match them to their corresponding states, allowing users to obtain a clear understanding of the origin of each state displayed in the Reactime UI as well as improving the overall readability
- Re-engineered D3 graph to display multiple state changes branched from the same snapshot through designing a new algorithm that recursively builds out a deeply nested hierarchical object with the given state snapshot data and optimizes hierarchy arrangement based on the input, giving users a clearer visual overview when switching between various state changes and improving the developer workflow
- Utilized Travis CI to achieve full Continuous Integration for the extension by configuring Docker-compose to enforce Jest tests to pass before any merge requests can be accepted into the master/dev branch, minimizing any errors and ensuring the code is production-ready
- Used Jest to accomplish complete unit and integration test coverage for the NPM package, streamlining the development process and maintaining code quality by applying common user usage patterns

**ForwardPMX** | SQL/Automation Engineer 2017 - 2019

- Developed a SQL query generator using React that supports dynamic horizontal and vertical table joining to speed up the data pulling process when working with company's proprietary relational database system and improve workflow for the media team
- Employed Puppeteer and Node to engineer browser automation APIs that automated manual behaviors while minimizing resource usage, effectively reducing the time required to complete different client requests for account managers

## Open Source Projects

**Server Visualizer** | NPM Library for Express Server Activity Monitoring

- Engineered an Express middleware, pre-packaged in an NPM package, that accumulates user's Express server activities and dynamically generates a new endpoint off of the server to serve up the React visualization, minimizing the user configuration required during setup
- Utilized Node's process module to extract instantaneous and differential timings from server activities, ensuring the reliability of the metrics
- Leveraged React and Chart.js to create a frontend architecture that parses through incoming data from the middleware and creates a set of intuitive, elaborate visualizations that displays real-time updates on the HTTP activities and status codes from user's server

**Time2Dungeon** | Online Multiplayer Card Game Inspired by "5-Minute Dungeon"

- Established WebSockets TCP connection protocol between the server and clients through a pub-sub architecture to synchronize game state across all clients, enabling real-time communication with minimal latency and ensuring consistency in the on-screen content among players
- Implemented OAuth2 to offload authentication to a popular and trusted third-party authentication solution, eliminating the need to store and hash sensitive user information manually and simplifying the registration process for the players
- Deployed Redux to create "one storage of truth" for all the in-game state, keeping the state flow predictable and minimizing prop-drilling

**Focus Today** | Multiuser Hybrid Productivity Tool

- Leveraged Bcrypt's salt hashing system and Express's session middleware to architect an authentication system that encrypts sensitive user information and prevents any cookie hijacking, resulting in improved security of sensitive user information and maintaining scalability
- Utilized React Router to create a single page application with dynamic client-side routing in order to minimize server calls and prevent APIs that have direct access to personal information from being exposed to unauthenticated users

## Public Talks

**Digest the Fiber: React Fiber Overview |** CSLabs **The What, How and Why of gRPC** | Build With Code NYC

## Education

**Baruch College** Relevant Course: Principles of Web Design
Awards: Baruch Entrepreneurial Intern Fellowship Recipient

## Hobbies

Music production | Following the latest smartphone news | Listening to J-Pop | Obsessing over Japanese variety shows