Chris Flannery | Software Engineer | New York City

973-897-5541 | chriswillsflannery@gmail.com | github.com/chriswillsflannery | linkedin.com/in/chriswillsflannery

TECHNICAL SKILLS

Strong: JavaScript (ES6), Webpack, React (Fiber & Concurrent), React Router, Redux, HTML, CSS, Sass/SCSS, Typescript, Node.js, Express, SQL Databases (PostgreSQL, MySQL), NoSQL Databases, GatsbyJS, OAuth, Bcrypt, Agile/Scrum, Git, Interpreters/Abstract Syntax Trees
Experienced: GraphQL, npm packaging, serverless, jQuery, React Context API, Accessibility, TDD (Jest, Enzyme, Supertest & Puppeteer), Shell/Bash, JWTs, Docker, AWS (EC2/S3/RDS/Lambda), TravisCI/Github Actions

EXPERIENCE

Reactime | *Open Source Product Developed and Maintained under OS Labs* | Software Engineer **2019-2020**

• Implemented Agile development process, facilitating code reviews and task management to lead engineering team through iteration on React time-travel state debugger, expanding beta support for Hooks, Context API and Routers by refactoring and broadening Fiber traversal algorithms.
• Enabled Concurrent Mode support by writing modularized tree-building code consistent with Concurrent React's asynchronous dual-tree diffing algorithm to introduce interruptible rendering, reducing the need for UI debouncing/throttling, and ensuring responsiveness at runtime.
• Added Suspense (React 16.8+) experimental feature support, following Fiber's underlying node tree to identify specific symbol-attributed components and appropriately discarding fiber segments as needed to implement lazy loading and declarative specification of loading states.
• Utilized Acorn to generate an Abstract Syntax Tree to capture users' hooks declaration hierarchies, following the thread of execution to dynamically retrieve hook types and match them to their corresponding states, strategically leveraging Acorn's size and performance advantages.
• Interfaced with Browser History API to dynamically capture session route history, enabling time-travel support for React Router configurations.
• Configured an event emitter to dispatch state changes from npm package, ensuring proper reconciliation between changes without side effects.
• Expanded Jest testing suite, using the Enzyme utility to provide a less opinionated yet more verbose environment for React component testing.
• Architected GatsbyJS marketing site, using serverless integration to reduce complexity in deployment to Github Pages, and relying on its efficient native compilation method to push data transformation to build-time to ensure elimination of any errors in syntax and semantics.
• Leveraged GraphQL to declaratively and dynamically import static assets and improve performance by eliminating data bloat and boilerplate.
• Used TravisCI to automate continuous deployment testing suite, creating a sophisticated deployment pipeline for further npm package updates.

Sackman Enterprises | Multimedia Developer **2018-2019**

• Utilized GUI to interface with MySQL-based CMS, leveraging its on-demand scalability for efficient management and customization of a deeply embedded application, and its security of transactions to ensure data integrity and ease of use for e-commerce site clients and maintainers.
• Customized CSS using fluid grid system to certify uniformity across multiple devices and provide the broadest user experience for a large base of diverse clients, reducing browser scaling errors, and providing greater accessibility, especially against socio-economic bandwidth restrictions.
• Used JavaScript to integrate transactional email service with SaaS, creating an intuitive way for non-developer client to store relational data.

Fuerza Strategy Group | Oversight of multimedia development, production and purchases for political strategy firm **2017-2018**

• Applied HTML, CSS & JavaScript to rapidly prototype and build political campaign websites and email templates built on responsive CMS.

OPEN SOURCE PROJECTS

CommSource | *Ecommerce Templating Engine* **2019**

• Built Webpack using webpack-dev-server with watch mode and hot reloading as a proxy to enable faster service of bundled content from RAM.
• Constructed NoSQL database favoring availability, partition tolerance and speed, using an ORM modeling rule to handle large amounts of data.
• Implemented CSRF token validation for user sessions to ensure valid HTTP requests and prevent against unintentional user action induction.

oAuthA | *Node Authentication Middleware* **2019**

• Integrated OAuth 2.0 and cookies by developing express middleware to decouple Google, Github, LinkedIn and Facebook protocol, simplifying the user authorization flow and offloading vulnerabilities by externalizing part of the authentication security to dedicated security providers.
• Employed Node.js using npm to build and publish package which combines multiple OAuth middleware into one set of logic, bootstrapping and optimizing the OAuth implementation for engineers, leveraging the V8 engine to provide extremely fast and performant compilation times.
• Used SuperAgent to write lightweight, progressive AJAX queries to streamline data fetching and for its highly configurable plugin ecosystem.

Split.io | *Powerlifting Routine Generator* **2019**

• Configured Node.js & Express-based server using MVC pattern to concurrently handle HTTP requests to multiple endpoints and architected hierarchy of routers and controllers, providing client access to persistent and scalable data flows and RDBMS-connected CRUD operations.
• Created SQL database using connection pooling to bind database to Node.js server, applying object-relational mapping and enforcing ACID compliance to foster data integrity and scale while emphasizing strict data typing and persisting user data with browser-based SSID cookies.
• Combined Bcrypt's hashing function with native and UUID4-based salting to securely store user info and provide resistance against brute-force rainbow and dictionary attacks, even with increased computation power, using Bcrypt's key factor feature to adjust the cost of hashing.
• Interfaced with cloud-based neural machine translation service to parse linguistic data and generate lightweight, readable, serializable output.

EDUCATION

The College of New Jersey | B.A. Interactive Multimedia

RECENT TALKS / ARTICLES

Typescript: Real World Development Implementations; Apache/nginx vs. Serverless - Build With Code NYC, 2019
Deep in the Weeds with Reactime, Concurrent React _fiberRoot, and Browser History Caching, ITNEXT, 2019

INTERESTS

Synthwave | Outlaw country | Chicken parm | Powerlifting | Thrifting (esp. leather) | DIY tour routing | Good whiskey | Grunge art