Lily Skye (@suchipi)

github/suchipi
npm/~suchipi
me@suchipi.com

- **JavaScript guru** with expertise in (but not limited to) **runtime semantics, syntax, and language specification**.
- **TypeScript expert** with deep knowledge on type-checking semantics, performance/maintainability tradeoffs.
- Enthusiastic about **JavaScript, TypeScript, Vitest/Jest, ASTs**, developer tooling, and developer experience.

Skills

- 10+ years of experience using **TypeScript, Babel, Webpack, Linux, Docker, Node.js, Jest/Vitest/Mocha, NPM**, and **GitHub** to meet the needs of rapidly evolving codebases.
- Former **collaborator/co-maintainer of Babel.**
- Former **core maintainer of the Prettier code formatter.**
- Contributor to **Jest**, **React DevTools**, and **core-js.**
- Creator and maintainer of Hex Engine, a **zeroconf TypeScript** game Engine with a **React-like** design and custom **Webpack Loaders**.
- **315 repositories on GitHub** (including 41 private repositories).
- **198 packages on NPM**. Has been publishing NPM packages for 9 years.

Expert at: JavaScript, TypeScript, Babel, Prettier, Webpack, Node.js, Jest, Cypress, Mocha, Jasmine, NPM, GitHub, Linux, Docker

Comfortable with: Java, Objective-C, Ruby, Swift, Lua, C#, ANSI C, Go, Selenium

Exposure to: Rust, OCaml, Reason, C++, Python, Harelang

Interest in Learning: Rust

Work History

- **Senior Software Engineer at ControlZee** - 2/2022 – 7/2025

  Contributed to the browser-based 3D multiplayer game platform dot big bang. Iteratively and sustainably **transformed a large ES5 codebase to modern TS/TSX**, designed and implemented a **versioning system for the user-facing game engine scripting API**, and maintained and added features to the user-facing **game engine scripting API and code editor**. Also set up and evangelized **unit testing and end-to-end testing** for the application, and made the application flexible enough to allow unit testing in Node.js.

  - Evaluated the company's failed past attempts to **modernize the codebase with Webpack and TypeScript**, and **planned and proposed an iterative approach** to achieve the same in a way that could be safely merged piece-by-piece with low risk of regressions. Upon putting the plan into effect, **achieved in 3 months what the company had previously been failing to do for 2 years**.
  - Built **custom tooling and codemods** related to the aforementioned conversion. Custom scripts tracked dependencies in a codebase that was **migrating from globals to ECMAScript Modules**, and **custom linting rules** helped identify and prevent patterns we were trying to move away from.
  - Built a **custom Docker image for the frontend** (webpack/typescript/eslint/etc) local development Docker Compose service, utilizing a **custom-authored Node.js application** which used filesystem watchers to intelligently restart webpack, TSC, and run npm install when certain files on disk changed, like package.json or webpack.config.js. The same tool also reacted to changes in git branch, allowing **developers to switch git branches as needed by their workflow** while ensuring that the **frontend services were always in the correct state**.
  - Designed, implemented, and maintained a **versioning system for the user-facing TypeScript-based game scripting API**, and created a corresponding process for adding to the API. Breaking changes were tracked using **semantic versioning**, and **each individual script in the same game could use a different version** of the API, and could talk to other scripts using different API versions. As the game engine matured, the corresponding user-facing **API objects kept their stable interface so user's games didn't need to be changed**. Educated coworkers on how the system worked and how to add to it.
  - Added **fully-typed ECMAScript module support (import/export) to the TypeScript-based game scripting API**, using a **custom-made implementation of CommonJS** that tied into the engine's in-memory models for games and user scripts. Added features to the **Monaco-based code editor** interface in order to support this feature.

    - Designed and maintained the **webpack config** for the project over the years, adding new features as requested, such as **bundle splitting** (which involved changes to deployment and dockerization), **custom loaders and plugins** for various vendored dependencies and git submodules, dead code elimination and dev-build-only conditionals, ability to run webpack in production mode in a local environment for the purposes of performance testing, and more.
    - Set up **Jest for unit testing and Cypress for end-to-end testing**. Instrumented different aspects of the engine and untangled cyclical module dependencies in order to make this all function. Implemented both in CI (Jenkins) using custom Docker images. The Cypress Docker image was created by hand. Both Docker images were used in both CI and local development. The Cypress Docker image exposed its X framebuffer display via VNC, and also served a browser-based VNC client for convenience. Made adjustments to the engine to use deterministic randomness when requested, in order to support image regression testing of running games.
    - Built a **fully-typed seedable random-number-generator API** for the user-facing game scripting API, based on requests from game developers. Supported **custom seeds, 1D-4D simplex noise, weighted draws, and more.** Many convenience methods were present for eg. picking an item from an Array or making a random string of a specified length.
    - Put several different **code health checks** into the codebase that would run **in a dev's editor and also in CI**. A **linter config with custom linter rules** was created to suit the needs of the evolving project, and custom shell scripts were built to **track module dependency cycles** and **disallow merging code which introduced new dependency cycles.**
    - **Assisted developers as needed** when they encountered issues with Docker and the like, made **changes to the Docker configs to avoid these issues** in the future, and led an effort to switch our macOS devs from **Docker Desktop to Orbstack**, after testing the latter on my own and determining it would be a better fit for us.
- **Voluntary time off from working, to spend with family** - 10/2021 – 2/2022
- **Senior Software Engineer at Webflow** - 10/2018 – 10/2021

  Created and maintained internal developer tools used company-wide by engineers, QA, product designers, data analysts, and management to test, deploy, analyze, compile, and iterate on code for the Webflow platform. Additionally, **maintained and improved configs and config management for Babel, ESLint, Webpack, Prettier, etc** to fit the needs of the entire engineering organization. Some highlights:

    - **Created a process management app with a visual dashboard** (written in TypeScript and rendered in the browser) which replaced a large, complex shell script responsible for launching all local development processes. Innovations and architectural improvements in **the new app allowed us to reduce CPU/RAM usage** on engineer machines significantly, **centralize config management** across processes in one unified interface, and **improve the user experience of understanding**, learning about, and debugging processes on your local machine.
    - Created and maintained an AST-based **module graph visualizer and browser** application that identified and explained hundreds of dependency cycles in the codebase. The engineering team used this tool to remove these dependency cycles and therefore fix and avoid bugs and race conditions.
    - Worked to iteratively convert a large monolithic Node.js app into a group of modularized pieces, by creating a **custom, opinionated monorepo workspace management system** built on top of open-source technologies. This effort contributed to better engineer onboarding, the ability to deploy pieces of Webflow more modularly, the ability to develop individual packages in different programming languages, and more.
    - Created tools that allowed us to write automated tests for CI scripts and local tooling code, including a **seedable virtual filesystem and virtual git repo system, which are both transparent to the code under test**. With these tools, I wrote automated tests for many CI scripts and local tooling code, identifying bugs and fixing them along the way.
    - Drastically improved and modularized the CI pipeline, decreasing runtime and improving the experience of debugging CI. Created **a clear pattern and system for performing arbitrary checks across modified files** in the codebase in both CI and locally, which allowed teams to codify their own specific high-level concerns in order to **maintain the health of the codebase and warn others about potential concerns**, automatically.
- **Nexia Home Intelligence** - 5/2015 – 10/2018 (3 Titles)
  - **Software Engineer at Nexia Home Intelligence** – 11/2016 – 10/2018

    Continued at Nexia as the lead engineer for their hybrid mobile web app and its associated backend/API. Codesigned and implemented a device enrollment procedure for devices with limited or no displays. Lead the mobile engineering team in decisions regarding the future growth of the codebase in order to speed up turnaround time for mobile work by 2-3x. Traveled to interface with engineers at partner companies to standardize API decisions and lead solutions. Gave several lunch-n-learn presentations to internal teams teaching how to use React, how to use and debug Flow, and strategies for solving problems in the IoT domain with React. Implemented novel workarounds and polyfills for layout, rendering, or behavioral bugs in mobile browsers using CSS, JavaScript, Java, Objective-C, and C#. Applied performance enhancements and code-splitting to reduce startup time of the mobile app from 8 seconds to 1.5 seconds. Implemented native-feeling animations, gestures, and user interactions within the limitations of a mobile web browser. Wrote a custom CI server to replace Jenkins to reduce time lost to debugging the CI pipeline.

  - **Developer/Analyst at Nexia Home** Intelligence – 10/2015 – 11/2016

    Continued work on a managed, IoT smart home system with a focus on its mobile hybrid web app frontend, targeting Android, iOS, Windows 10 Mobile, and desktop web browsers. Learned Java, Objective-C, and C#, and became intimately familiar with JavaScript. Led and completed an effort to convert the frontend of the mobile app from Backbone.js and Polymer to React. Became a core engineering team member for the mobile app and conducted phone screening, interviews, and paired programming exercises to find and fill positions in the mobile engineering team. Mentored new employees working across the stack and taught a contractor with no prior coding experience how to develop full-stack for the mobile application. Set up and managed a CI server (Jenkins) and automated mobile app builds and deployment, and streamlined app delivery for testing.
    Used Docker to run backend services locally and debug platform-specific issues.

  - **Contractor at Nexia Home Intelligence** – 5/2015 – 10/2015

    Developed full-stack cloud-to-cloud API solutions for a managed, IoT smart home backend and its hybrid mobile web app and web frontend. Used Ruby on Rails, Backbone.js, jQuery, and CSS. Worked with a System Test team and stakeholders to verify implemented work. Participated in team architectural discussions to solve infrastructure problems.

- **Self-Taught CS Student** - 2/2015 – 5/2015

  Took time off from working and spent 3 months teaching myself how to code full-time, 7 days a week. Learned Ruby on Rails, Git, JavaScript, HTML, and CSS. Learned via courses on Codeschool.com and reading the docs of the technologies I was learning. Built a multi-user image browsing, querying, and indexing application as I learned.

- **Logistics Team Member at Target** - 9/2014 – 2/2015
- **Founder of The Nerd Herd (Computer Repair Company)** - 6/2013 – 9/2014
- **IT Intern at Adams 12 Five Star Schools** - 4/2011 – 6/2013. Age 15 at start of internship.

Education

Self-taught software engineer with ~7 years of experience, including years of open source software collaboration.

Graduated from Legacy High School in 2012, including Computing, Networking, and AutoCAD courses at Bollman Technical Education Center.

Received first-place at the 2011 SkillsUSA competition in Computer Repair at the District and State levels.

Hobbies

Final Fantasy XIV, game modding, 3D printing and design, open source software, tinkering with laptops and mobile devices, PC building, community fostering, learning new things, trying new foods, and more.