

Paul Dill
Front-end Software Engineer
Portfolio: <https://pdill.dev>

paul@pdill.dev
808-464-3859

Professional Summary:

Self taught software engineer most recently employed in the manufacturing controls industry. I've been furloughed for a few months between projects, so I'm applying to jobs to find a great new opportunity for growth. I'm eager to learn and grow with a good company. Willing to learn any programming language, and have a proven track record of picking up new skills and information quickly. My method for developing programming skills has been based on reading documentation, building personal projects, prototypes, and making open source contributions.

Technical Skills:

- | | | |
|------------------|-------------------|-------------|
| • Angular | • SASS | • Bash |
| • React.js | • Material Design | • Yarn |
| • TypeScript | • HTML5 | • NPM |
| • ES6/JavaScript | • REST API | • Git |
| • Next.js | • Node.js | • Gitlab |
| • Vercel | • MariaDB | • Github |
| • Cypress | • Ubuntu | • WordPress |
| • CSS3 | • Windows 10 | |

Relevant Work Experience:

KS Controls Inc.: Software Engineer

Greenville, SC / Mintraching, Germany / Captain Cook, HI (remote)

- Automated logistics and manufacturing solutions.
- Spearheaded the design of a mobile version of their iCMS program which now allows manufacturers to use this multi-million dollar product from their own devices.
- Create user friendly UI components for an integrated control system using Angular, TypeScript, D3.js, Rx.js, and SASS.
- Accomplished roughly 80% test coverage of the iCMS system using Cypress and Jasmine which contributed to a more complete CI/CD pipeline which ensures that sneaky bugs aren't being added to the codebase.
- Utilize object oriented design pattern to create modularized, extensible classes.
- Consume data via SignalR subscriptions to give users real time updates about manufacturing machines.
- Utilize dependency injection to create decoupled and easily update data providers.
- Reduced codebase size by creating generic components to handle commonly repeated logic. This allowed us to make rapid changes to the front-end as the back-end API changed saving the company ample time and money.
- Document code using comments, semantic commits and Gitlab issue tracking so that the entire team can easily figure out how, when and where code changes were implemented.
- Keep open communication between Germany, South Carolina, and Hawaii with front-end team, back-end counterparts, and project managers to ensure that customer expectations are being met.
- Work independently on a remote team across 12 time zones.