James Waples

Cambridge, United Kingdom
07714 774 508
james@wapl.es
https://wapl.es
https://github.com/jamwaffles

Technical Skills

Likes: rust optimization git json postgresql reactjs typescript docker sass embedded
Dislikes: java basic svn php angularjs angular

Experience

Full stack developer – Fluidic Analytics *Mar 2020 → Current*
rust, postgresql, event-sourcing, typescript, reactjs

- Work with another developer to architect and implement the company's new cloud application from scratch using a Rust backend and Typescript/React frontend. We use Jira to gather and track requirements and progress.
- Ensure the application is compliance-ready using an event sourcing paradigm to record the granular history of actions against entities in the application. What PII and where it is stored is also taken into consideration to comply with the GDPR.
- Release the event sourcing machinery as an open source library for reuse in other applications and projects.
- Experiment and implement a granular but not too complex permission system for both the backend and frontend to ensure security but also clearly message to the user what they can and cannot do.

Senior/lead developer – Repositive Ltd *May 2018 → Mar 2020*
typescript, rabbitmq, webpack, microservices, reactjs, redux, rust, bioinformatics, docker, google-cloud-platform, kubernetes, sass, bulma, circleci, actix-web

- Maintained and refactored a microservice-based backend in Typescript for our main product as part of the development team.
- Handled development and maintenance of features for the company's React-based commercial website, with a focus on speed and SEO.
- Maintained and developed a Typescript library to make using the event-sourcing paradigm easier for the dev team, learning to carefully consider the API surfaces and changes when bumping semver versions.
- Evangelised Rust within the company, producing some data-processing tools to work through large amounts of bioinformatics data safely and quickly using multithreaded code where possible.
- Scratch-built a brand new platform after a pivot, using Rust with PostgreSQL as the backend API, and Typescript with React and Redux in the frontend.
- Work closely with product manager to wireframe, scope and design new features for the platform.
- Worked with recruiters to hire new developers into the team when funding allowed expansion.

Developer – Cambridge Intelligence *Sep 2017 → May 2018*
javascript, circleci, amazon-web-services, amazon-ecs, docker, docker-compose, mocha, node.js, webpack, elm

I'm a JavaScript developer working primarily in vanilla JS, but also with React and various JavaScript tooling such as Webpack and Mocha.

My main responsibilities at Cambridge Intelligence include:

- Implement demos to help users work with the core KeyLines product.
- Help others use AWS to maintain infrastructure and perform day to day tasks and debugging.
- Lead development on an in-house, Docker and ECS-based tool to automatically host multiple in-development branches so new features can be tested in reproducible, isolated environments. Responsibilities include writing and reviewing React frontend code, designing and maintaining AWS infrastructure.
- Reduce tech debt and make development faster and less painful for me and other developers on my team. This involved moving from LambCI to CircleCI, reducing build times and build system complexity and refactoring the codebase to use ES6 native modules instead of global variables.

Developer – Media Ingenuity / Totally Money *Jul 2014 → Sep 2017*
reactjs, node.js, javascript, gulp, osx, redux, nginx, postgresql, rust, babeljs, webpack, amazon-web-services

Main responsibilities and tasks:

- Architect and lead frontend development of the TM Free Eligibility Report, involving a universally rendered React/Redux application with authentication which integrates with a .NET backend API.
- Architect and maintain the TM Loans frontend webapp, again written with NodeJS and React. It originally used Flux, however I refactored it to use Redux to reduce the levels of technical debt present in the codebase.
- An internal monitoring and toolbox application involving fetching data from various AWS APIs and displaying charts of data using D3. It also uses websockets for realtime communication. I also took this as an opportunity to see how Rust performs in production with a small log file parser run as part of the infrastructure.

Mobile and ecommerce web developer – Long Tall Sally *Jun 2013 → Jul 2014*
php, javascript, mysql, gruntjs, less, jquery, backbone.js, hybris, marionette, node.js, slim, redbean, c3js, gulp

Main responsibilities and tech:

- Ecommerce website maintenance and development (PHP, MySQL, LESS, JavaScript)
- Maintenance and development of in-house admin and reporting tools (PHP, MySQL, LESS, JavaScript)
- Writing awesome new reporting tools designed for speed, good looks and usability (NodeJS, React, MySQL, LESS, JavaScript)
- General bug-fixing for all our repositories
- Trying really hard to turn all our SpaghettiPHP into something more easily workable.
- Monitoring, maintaining and creating servers with Rackspace VPS (Linux, Nginx, Ubuntu)

My role at Long Tall Sally started with the redesign and redevelopment of our ecommerce site. I implemented the frontend using Backbone with Marionette, LESS for styles and Handlebars for templates, all built with a Grunt toolchain. The backend was upgraded from a horrid legacy system into a more modern architecture utilising some aspects of MVC. (It was difficult to port the system entirely as it would've required a complete rewrite.)

I also wrote and maintain(ed) an in-house administration system written in PHP (Slim framework with Redbean 3.x ORM) using Twitter Bootstrap on the front end for fast development.

Another project involved fetching historical data from a weather API and displaying a comparison between now and last year for our store locations. I used NodeJS with Sequelize and Q (promise library) on the server, and Backbone with ReactJS in the browser.

One of the features on longtallsally.com required an insights/analytics page so members of the marketing team could monitor user feedback. I wrote an application using NodeJS interfacing with a MySQL database. To render pages, I decided to use React to render the same view both on the server and on the client (isomorphic programming). It was fun seeing the page load pre-rendered, but being completely interactive without having to add any extra code.

Web Developer – David Henderson of CliqueSoft.org (DigitalPipe) *2010 → 2013*
php, html, javascript, mysql

I work one-to-one as a freelance developer for fixed fees to create websites and web applications in my spare time. My jobs include the following for single- and multi-page websites:

- Designing the user interface and UX
- Developing the frontend logic in JavaScript
- Developing the backend system

Design of the interfaces usually begins with a mockup from my client exhibiting the general concept. I then create a more refined mockup (in HTML) of the design, which is checked and changed by the client. When the mockup is complete, front- and back-end development begins.

The backend is implemented in PHP and MySQL served by Apache and is object oriented in design to keep the code DRY and concise. The frontend logic is implemented in jQuery to make JavaScript development faster.

Web Developer – Cromwell Business Systems *Aug 2012 → Jun 2013*
php, mysql, apache, css, jquery

I was part of a team of web developers that manage our customers' websites. Some examples of my previous projects include integrating Bazaarvoice reviews into a client website, updating and bugfixing a customer's website and web support. My other work involves:

- Fixing bugs that arise from user testing and live usage
- Adding features requested by customers
- Maintaining the web software stack (LAMP)
- Improving the codebase

The web stack runs on PHP, MySQL and is served by Apache. I frequently manage all 3 technologies as part of my work.

All sites are based on a custom-written framework running on PHP 5.1.6. I spent some time optimising and cleaning up the codebase to improve customer experience. This was achieved by profiling runs with XDebug, and benchmarking on the client using Chrome's developer tools.

Website Redevelopment – Zing Conferences *2011 → 2011*
php, mysql, css, jquery, html

My task was to redesign and redevelop the company's existing website which was old, outdated and buggy. The staff of the company used the website as a management system, however the workflow was suboptimal.

The brief was to:

- Create a new design for the website matching the company's rebranding
- Add features to the public-facing website to make it easier to use
- Implement an extensive backend management system to maintain conferences and attendees as efficiently as possible

Internal IT support – Dept. of Earth Sciences, Cambridge University *2010 → 2010*
linux, ssh, windows

I joined the team at the Department of Earth Sciences in Cambridge for two weeks of work experience. I was trained in basic system administration, and learned how the network infrastructure functioned. I also responded to technical support calls from users of the network.

One of my main tasks was the remote administration of Linux machines, as well as printers and other hardware on the network.

The Earth Sciences Department also maintains a small museum. While I was there, a new exhibit was being built with an interactive Earth projecting as it's centerpiece. I was part of a team that built, set up and configured the interactive globe for use by visitors to the museum.

Education

Level 1 MIG Welding – Cambridge Regional College *Aug 2016 → Oct 2016*

I did a 10 week evening class and learned how to weld using MIG technology. I also tried a bit of TIG and stick welding which was very fun.

A Level Computing – Hills Road Sixth Form College *2010 → 2012*
pascal, html, css, javascript, php, mysql

Over the 2 year course, I filled in the missing parts of the theoretical side of my knowledge. At the end of each year, I took two exams with the second year also requiring a piece of written coursework.

The coursework was to consist of the documentation for a programming project, including printed source code. I had acquired some freelance work at the time, so I chose to use that project (named Threads) as the basis for my coursework.

A Level Electronics – Hills Road Sixth Form College *2010 → 2012*
avr-gcc, embedded, c

I achieved grade A in A level electronics.

The course involved two written exams and 3 practical pieces of coursework. Each piece of coursework consisted of a circuit and a writeup. The circuit was designed, built and tested by each student and a writeup describing the process handed in along with photographs of the finished circuit.

My circuit was a 3 wire serial link using (mostly) discrete logic. The only non-discrete part was an ATMega32U2 with code written in C acting as a parallel to serial encoder.

GCSE Electronics – Impington Village College *2007 → 2007*
electronics

I achieved grade A* in GCSE electronics, two years early in year 9 (as opposed to year 11)

The course was mainly self-taught, with an hour each week with a teacher to guide me through the textbooks and to offer help and suggestions for the practical side of the coursework.

My final piece was a portfolio documenting the design, implementation and testing of a simple alarm system.

Projects & Interests

Stack Overflow – https://stackoverflow.com/users/383609/bojangles *Jul 2010 → Current*

Written 651 answers. Active in arrays, css, html, image, javascript and 17 other tags.

Rust LinuxCNC HAL – https://crates.io/crates/linuxcnc-hal/ *Jan 2020 → Current*
rust, cnc

A Rust interface to the hardware abstraction layer (HAL) of the popular LinuxCNC open source machine controller.

Aims to allow user components to be written in Rust to make them more robust and pleasant to hack on than the traditional C components normally written for LinuxCNC.

Embedded Graphics – https://github.com/jamwaffles/embedded-graphics *Feb 2018 → Current*
rust, embedded

A general purpose, low memory footprint graphics library aimed at embedded systems, but usable anywhere.

jamwaffles/ssd1306 – https://github.com/jamwaffles/ssd1306 *Feb 2018 → Current*
rust, embedded

Small OLED display module driver for a 32 bit ARM microcontroller, written in Rust.

Public Artifacts

Optimising Rust: Clockwise Triangles – https://wapl.es/rust/2020/07/25/optimising-with-cmp-and-ordering.html *Jul 2020*
rust, rust-no-std

Optimising stuff for embedded devices is hard! This post goes through some of the process and evolution of an integer-only solution to computing line joints for thick polylines in embedded-graphics.

Cross compiling Rust from Linux to macOS – https://wapl.es/rust/2019/02/17/rust-cross-compile-linux-to-macos.html *Feb 2019*
rust, cross-platform

I've recently been working on a Rust project at work which requires compiling for Linux (GNU), Linux (musl - for Alpine Linux) and macOS. This post collects snippets and tips from around the web to walk through creating a working cross compilation from Linux to macOS (among other targets)

Spindle speed control using LinuxCNC 2.7 with a Huanyang inverter – http://waple.es/cnc/2015/12/04/huanyang-linuxcnc-2.7-speed-control.html *Dec 2015*
cnc, linux

Documenting the lengthy, confusing process of getting a Chinese made 3 phase speed controller working with LinuxCNC.

Readings

Pro Git (Expert's Voice in Software Development) – Scott Chacon – http://www.amazon.com/Pro-Experts-Voice-Software-Development/dp/1430218339%3FSubscriptionId%3DAKIAIIBINOD46VC3JCLQ%26tag%3Dstackoverfl08-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D1430218339

Git is an incredibly powerful version control system. I use the online version of this book as a reference on the more advanced parts of Git.

Coding Horror: The Future of Markdown – http://www.codinghorror.com/blog/2012/10/the-future-of-markdown.html

Markdown is a simple little humane markup language based on time-tested plain text conventions from the last 40 years of computing.

Virtues of a Programmer | Terminally Incoherent – http://www.terminally-incoherent.com/blog/2013/04/17/virtues-of-a-programmer/

Larry Wall once said that the three virtues of a programmer are Laziness, Impatience and Hubris. This saying has become legendary tidbit of software development humor, but Hubris and cowboy programmers are a terrible vice.

PHP: a fractal of bad design – http://eev.ee/blog/2012/04/09/php-a-fractal-of-bad-design/

An objective, quantitative list of why PHP is bad and one should use any other language to write webapps.

Cargo: predictable dependency management – http://blog.rust-lang.org/2016/05/05/cargo-pillars.html

A really interesting blog post about the virtues and workings of Rust's package manager, Cargo.

What Elm and Rust Teach Us About the Future – Martin Černý – https://dev.to/martincerny/what-elm-and-rust-teach-us-about-the-future

Rust and Elm bring several new, interesting ideas to the programming language world. This is a good article discussing the merits and some of the disadvantages of both languages.

Tools

First Computer:	An old Windows 98 box salvaged from my primary school
Favorite Editor:	Sublime Text 3