

Félix Saparelli

WHAT I KNOW: **Laravel 5, PHP 5 and 7, lots of JavaScript, Linux admin, Rust.**

I've been writing and reading PHP, Ruby, and JavaScript for nigh on a decade, although it has only been the last six years (with a two year break within that for studies) that I have done so professionally. I have been running Linux personally, for others, on servers, and inside containers for almost as long.

I like to *understand* languages and patterns to be able both to wield them better and to construct others building upon the knowledge. I like to explore and create, but I also love programs that work well and are built just right enough they don't need maintenance, and can happily keep running for years without intervention.

PROFESSIONAL EXPERIENCE

McKay Group, Whangarei — *PHP / Node.js / Docker*

LATE 2014 - PRESENT

Initially a frontend developer, I expanded into a **variety of roles**, ranging from **frontend and design work** to writing **application servers in PHP7, Node.js, Ruby**, to **managing clusters of Linux servers** and the services running within. I **designed entire systems** and took **ownership of projects** to lead them to completion and stability.

During my latest project, I started by creating a prototype for a marine industry client, then architected a scalable solution, built it along with a small team, deployed it to a flexible cluster, and managed load as it grew 1200% over a year.

NoCowboys, Auckland — *PHP*

2012

Took on a **large, difficult legacy CRM codebase** built with an old version of a proprietary PHP framework, and **refactored core components** to be more performant, documented, and overall easier to maintain and upgrade to keep up with evolving business requirements.

Changery, Auckland — *Ruby / JS*

2011

A (since-defunct) startup in the fitness / self-help industry. Heavily **test-driven development**. I built an application from the ground up with **Ruby on Rails**, Postgres, and jQuery. I was also involved in some product and graphical design.

Adventure Forest, Whangarei — *PHP / jQuery*

2008 - 2010

I designed and implemented the first website for the business, with only minimal aid from [PHP libraries](#) and jQuery. I then **added a custom CMS**.

CONTACT

Whangarei

Northland

022 601 3595

felix@passcod.name

Github: @passcod

LARAVEL

I have now built three large applications with **Laravel 5**, and a spat of smaller ones. While I would not call myself a specialist, I know my way around and can figure out the rest.

DATABASES

I have written for and managed small **MongoDB**, **Couch**, and **RethinkDB** databases, as well as larger **MySQL** and **Postgres** data systems.

SERVERS

I have set up, upgraded, and maintained **medium-scale service orchestration** (Systemd, Docker, and/or Kubernetes) over a range of Ubuntu and CoreOS servers.

LOCATION

I live in Whangarei, where I grew up after my family sailed here, and which I love.

LANGUAGES

I speak both **French** and **English** fluently.

EDUCATION

NorthTec, Whangarei — *Architecture Technology*

EARLY 2013 – LATE 2014

Essentially architecture but with less art and more technical details. Besides the skills handling **AutoCAD and drawing tools**, I gained an understanding of **design and artistic principles**, through the study of spaces and colour. On top of that, I also got practice **managing complex projects** that cover a myriad details involving small teams.

OPEN SOURCE PROJECTS

Notify — *Filesystem notifications for the Rust ecosystem.*

<https://github.com/passcod/notify>

LATE 2014 – PRESENT

Cross-platform filesystem notifications (for monitoring when files and directory structures change) abstracting several backends into a unified interface. Many contributors, **289 stars on Github**; a dependency of many projects, 55k downloads so far.

Cargo Watch and Catflap — *Project development tools.*

<https://github.com/passcod/cargo-watch> and <https://github.com/passcod/catflap>

LATE 2014 – PRESENT and MID 2017

Two **development tools** for the Rust/Cargo ecosystem, the latter spawned from a feature request on the former. **Cargo Watch triggers recompilation** or other tasks on project changes, including process restart, while Catflap provides **persistent socket binding** for TCP servers to avoid request-dropping and port conflicts.

DynWorker — *WebWorker management library.*

<https://passcod.name/DynWorker>

2011

Early in WebWorker history (back when Firefox was still in single-digit versions), it wasn't possible to load workers using **data:URLs** (you had to specify a script on the remote server). DynWorker was created to solve this, and added a **jQuery-inspired tool** to manage groups of workers and run off-thread computation **per-function rather than per-script**.