

# Félix Saparelli

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 systems so that I am able both to wield them better and to build others using those understandings. I like to explore and create, but I also get satisfaction from 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 PHP / JS developer, I expanded into a **variety of roles**, ranging from **front-end and design work** to writing **application servers in PHP7, Node.js**, Ruby, to **managing clusters of Linux** (Ubuntu, CoreOS; Docker Cloud, Kubernetes, bare) servers and the services (**databases, caches, discovery, storage, load-balancing**) running within to support the applications we launched there. I **designed entire systems** and took **ownership of projects** to lead them to completion and stability.

### NoCowboys, Auckland — PHP

2012

I was charged to **maintain and evolve their CRM** and public-facing website alongside the lead developer and founder while he was in Europe. The system was (at the time) built from a then-old version of a proprietary PHP framework, which added a challenge to an already *odd* codebase.

### 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

Designed and implemented the first website for the business, with only minimal aid from [Flourishlib](#), jQuery, and [PHP.js](#) (before I knew what a terrible idea that was). Also **created a custom CMS** for the less technical.

Whangarei  
Northland  
022 XXX XXXX  
[xxxxx@passcod.name](mailto:xxxxx@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.

## 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**.