
resume Documentation

Release 1.0.0

Chris McDonald

April 26, 2012

CONTENTS

1	Table of Contents	3
1.1	About Me	3
1.2	Technical Skills	3
1.3	Other Skills	6
1.4	Projects	6
1.5	Gigs	8

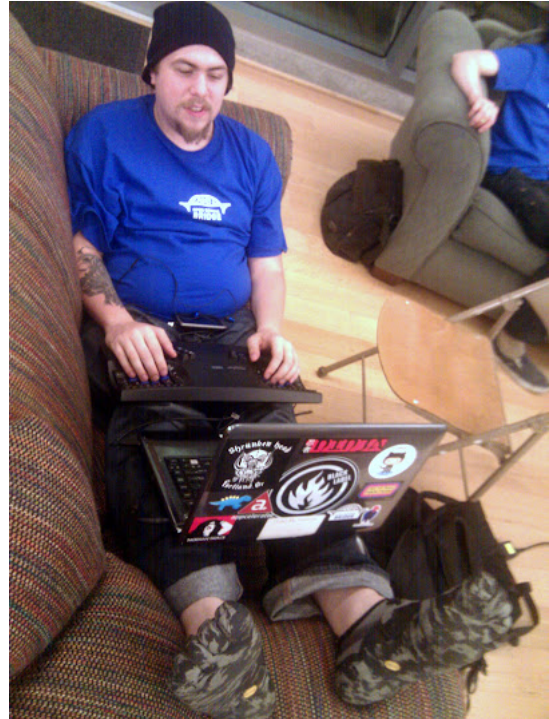


Figure 1: Open Source Bridge 2011.

Hi there, I'm Chris McDonald but most people call me [Wraithan](#). This is my web resume. If you are looking for a more concise one, you can find that [here](#).

Who is this guy?

- A polyglot who enjoys the commonalities and contrasts of each programming language and platform.
- A native of Portland, OR, USA. (And is not looking to relocate)
- A nerd, cyclist, and metal head.

But more than that I've been hacking in Python for 4-5 years at this point, and was primarily using it for web development and devops purposes, 3 of those years I was using Django.

I commonly am teaching myself new languages and frameworks in my spare time, most recently I began using Clojure which I find to be a lot of fun. I recently heard in one of the keynotes at PyCon this wonderful blurb: "When programming stops being fun, I'll stop doing it." and I think it describes how I feel about it as well.

TABLE OF CONTENTS

1.1 About Me

I am a social hacker from Portland, OR, USA. A large number of my evenings are spent hanging out with fellow programmers at cafes or bars. Commonly discussing our latest and greatest projects over some nachos.

1.1.1 Why do I hack

Repetition is one of the banes of my existence. The first time I do something it is new and interesting. The second time I get a feel for what parts should, could, and probably will be automated. The third time I am definitely looking for and/or writing something to solve this problem for me.

There is a lot of great software out there, functionality wise if not code beauty wise. I find over time instead of building my own unique projects that provide something that stands on its own, I am gluing services together in order to make the web a more coherent place.

Finally there is the wonderful micro high that comes along with solving a problem. It is even better if it was a hard problem or you have an elegant solution. The best is a really hard problem, really elegant solution, and being able to show it off to my friends because it is open source.

1.1.2 Personality

- Programming Polyglot
- IRC is one of the first things I reconnect to when I open my laptop.
- Karaoke enjoyer (unfortunately for my friends)
- Long time Linux nerd (I used slackware when it was cool)
- Cyclist, both commute and sport.

1.2 Technical Skills

This is a list of my stronger technical skills. I've played with writing things ranging from graphics engines, to decoding game save file formats, to window managers, to IRC bots, to web sites.

For a concise list of languages and time spent in them:

- 5+ years experience with Python, JavaScript, and SQL.

- 2+ years experience with C++, Perl, PHP, Java, and Lua.
- 1+ years experience with Haskell, C, and C#.
- Have played with many others including Common Lisp, Ruby, Go, and Clojure

1.2.1 Python

I've spent the last 5 years programming in Python, 3 of that was professional. Most of my experience with Python is centered around writing Django applications.

Some things I've built outside of web applications:

- A command line task management system
- Screen scrapers
- Feed aggregators
- Plugins and extensions for various tools that embed Python.

1.2.2 Django

If I am going to write a web app, I am probably going to start with `django-admin.py startproject <project name>`. I've been using it for years now on projects both large and small.

Here are some highlights of from Django projects I've worked on.

- Three sites that shared the same code base and served a large number of users.
- OAuth2 (spec 10 and 11) based signup and authentication.
- Upgrading between multiple Django versions.
- Numerous community and small business sites.

1.2.3 Celery

I use this when I need to offload tasks in Django based sites. Here are a few things I have done with it:

- Helped architect and develop a lazy caching backend that updated itself out of band using Celery, or calculated in line if celery hadn't updated the cache yet.
- Divided tasks into separated queues so the Celery daemon could be shared to multiple servers.

1.2.4 Fabric

This tool has saved me hours, if not days of my life.

- I have ran 2 sprints on it, one PSF sponsored, the other at PyCon.
- Made deployment simple and very reproducible causing it to be fast and take care of all the repetitive details for the team.

1.2.5 Redis

I reach for this when I want a key/value store or centralized pub/sub. I have use it for:

- Django caching backend.
- Django session storage.
- Celery queue backend (including connection pooling)
- Micro services based IRC bot using Redis' pub/sub as a transport.

1.2.6 PostgreSQL

This is my preferred relational database. It scales pretty well, it is open source, and I've come to rely on it anytime I need a database.

- Have used tools such as pgfouine to profile and optimize usage.
- Used pgbouncer to do connection pooling to decrease latency.
- Have scaled to tables with millions of rows.

1.2.7 JavaScript

I've used it for many years now. Mostly doing front end work on the web. But more recently I've also done things like building a Firefox add-on, and many little micro-services.

1.2.8 jQuery

When I am doing JavaScript for the sake of front end development I tend to lean on this library quite a bit for its selectors and other niceties.

- Built many dynamic front ends using AJAX
- Built a Firefox Add-on that uses jQuery to build and modify most of the DOM.

1.2.9 Firefox Add-on SDK

I've only built one but plan on building more.

- An add-on for listing GitHub repos and quick links for them (code, issues, wiki, etc)

1.2.10 Node

Node is a recent addition to my bag of tricks. It sparked the fire in me to really start enjoying JavaScript. I have been using it to build out co-operative micro-services such as:

- An IRC bot.
- A layer for receiving web hooks.
- A GitHub post receive hook processor.
- Process management for all of these micro-services.

1.2.11 Git

I am commonly found teaching people how to use git, recover from situations they are not sure how to get out of, and giving my opinions on best practices based on experience and discussion with others that have passion about how to use their version control system.

1.3 Other Skills

A listing of my less technical skills and development methodologies that I wasn't sure where else to put.

1.3.1 Agile

I have never worked anywhere that implemented any of the agile methodologies fully. But I have worked with several of the techniques found in Scrum and XP, including:

- Pair programming
- Continuous integration
- Sprint based development
- Test driven development
- Collecting and writing user stories

1.3.2 Public Speaking

Public speaking, especially educating a group on a topic is a lot of fun to me. Here is a list of places I've spoken or ran sessions

- PDX Python (Most recent to oldest)
 - Building Distributed Systems with Redis and Pub/Sub
 - django-slow-log
 - inspect (lightning talk)
 - dis (lightning talk)
 - baker (lightning talk)
 - Using bpython
- Barcamp Portland 2012
 - Co-ran a session on building distributed systems had people writing services for ZenIRCBot in the session to show how easy it is.

1.4 Projects

The vast majority of my projects are open source and can be found on [GitHub](#). This is a list of projects I've written or contributed to in some way. I love discussing them, so feel free to ask me about them.

1.4.1 Read the Docs

[Read the Docs](#) is a site for building and hosting [sphinx](#) documentation. The main goal of it is to lower the barrier to writing docs as much as possible. The idea is that if there is free hosting, automated building, and easy to select themes developers would write docs. Once they are written maintaining them is easy because when you push your docs are automatically rebuilt.

What I did for the project:

- Wrote better integration with GitHub, including tests.
- Made it possible for multiple people be admins on a project.
- Took part in architecture discussions with the maintainer.

1.4.2 ZenIRCBot

[ZenIRCBot](#) is a IRC bot that works a bit differently than your standard bot. Features (and interesting to implement things) include:

- Microservice architecture
- Redis pub/sub as a transport
- Services can be written in any language.
- Core bot written in Node.js but reference implementations are also in Python and Clojure.
- Patched various third party libraries to enable features in each of the bots.

1.4.3 GitHub Repo Widget

[GitHub Repo Widget](#) is a Firefox add-on that I wrote because I was on bad internet and wanted faster access to my issue tracker.

- Used the Jetpack Add-on SDK which is written in JavaScript.
- Wrote my own lib for interacting with GitHub
- Found and submitted several bug reports against addons.mozilla.org and the add-on SDK.

1.4.4 geoloqi-workout-tracker

[geoloqi-workout-tracker](#) is my stab at building something on the [Geoloqi](#) platform. Interesting bits of this:

- Register new accounts and login using Geoloqi's OAuth2 service.
- Associate accounts with DailyMile using OAuth2.

1.4.5 webhook-relay

[webhook-relay](#) is a service I was architecting to fill a hole in gluing webhooks together as well as making webhooks more robust.

- Forward and multiplex incoming webhooks using celery.

1.4.6 aichallenge-ants

This was [my entry](#) into [aichallenge](#) for the ants competition.

- Wrote a custom weight dispersion algorithm.
- Used PyPy.

1.4.7 speck

[speck](#) command line interface to [letsfreckle](#).

- Used [baker](#) to do the command line argument and function parsing easier.
- Interacted with an external REST API.

1.4.8 Fabric

[Fabric](#) is a library to make remote system management easier. While I've not directly contributed to Fabric, I have run multiple code sprints for it.

1.4.9 Resume

That would be [this](#). I have it up on GitHub because it is easier to maintain there. It is written in ReStructured Text using sphinx so I can host it on Read the Docs. This way I get a clean looking resume that I can export as a PDF, or just directly link to.

I was inspired by another user who has his [resume](#) on Read The Docs as well!

1.5 Gigs

My professional work in the software industry over the years. There is a bit of time between jobs here and there where I was working non-technical jobs and spending a lot of time on personal [projects](#) working to develop the skills needed to work in the industry.

1.5.1 Aquameta

Senior Software Developer (Mar '10 - Feb '12)

This was my most recent job, I loved this company and learned a great deal while I was working there. It is where I cut my teeth on Django apps that needed more than just some more hardware thrown at them to scale.

- Was part of a team that implemented, maintained, and extended a large Django application that powered 3 sites.
- Scaled that application using celery for offloading and generous amounts of caching.
- Upgraded Django between major versions twice. From 1.1 to 1.2 and from 1.2 to 1.3.
- Wrote and encouraged the writing of both unit tests and functional tests.
- Wrote and maintained one click deployment scripts using Fabric.
- Interfaced with the clients regularly to gather requirements for features.
- Guided the architecture of the application using community best practices and past experience.

1.5.2 Parthenon Software

Software Developer (Sept '09 - Nov '09)

This was a PHP shop I worked at for a short while. I did development on a well established code base.

- Updated unit tests, allowing for more confidence that application was correct.
- Met with clients to discuss and advise on what course to take for re-designing their software.
- Implemented feature requests, fixing existing bugs in the module while adding the feature, resulting in cleaner, better documented code.
- Participated in “brainstorming” sessions concerning design/testing details for various project.

1.5.3 Critical Path Software

QA Tester (May '08 - Aug '08)

Here I worked in the QA department testing software and hardware. The primary project I was hired for were 2 lines of computers that a company was going to release and they wanted some independant stress testing done in a wide range of activities. Online gaming, word processing, downloading content, watching HD video both streaming and off a Blu-Ray.

- Learned to write very effective bug reports.
- Wrote and executed test plans, tracking progress and reporting defects.
- Worked with a team to decide on software milestones and requirements.
- Set up many different hardware/software configurations for testing.
- Wrote a tool using C++ to generate data for testing.
- Assist in delegation of various portions of testing to help train new members of the team prior to product release.

1.5.4 Transim Technology

Intern Software Developer (Dec '05 - Aug '06)

This was my first foray into the world of software development at a company. The stack was a large java backend with a PHP layer on top with liberal use of Perl as glue.

- Cleaned up and maintained several in-house tools written in Perl, Java, and PHP for processing and displaying circuit schematics.
- Created a GUI for two of the in-house tools so that non-technical staff could assist in processing schematics that needed human interaction.
- Implemented a secure login system with detailed permission setup.
- Documented all of the above mentioned work, along with a large portion of a Java based webserver back-end.

I had a great time at this job and this, on top of my passion I already had, really sealed the deal as far as my desire to pursue software development as my career.