resume Documentation

Release 0.1

Randall Degges

Sep 27, 2017

Contents

Hi there, I'm Randall. This is my resume. If you'd like to see the **technical** details, you'll need to check out my resume project on github.

Note: My resume is currently *under development*. I'll remove this notice once it reaches a stable release ;)

In a nutshell,

- I'm a python / Django / telephony developer.
- I live in Los Angeles with my awesome wife Sami and dog Scribbles.
- I love geek humor, loud music, tech books, writing, and open source software.

CHAPTER 1

Table of Contents

About Me

I'm your typical hacker–I love learning new things, playing around with the latest tech, and figuring out how things work.

My passion is building software that makes things simple. There's something about the process of solving problems in an elegant way that excites me.

Philosophy

My philosophy on software and life is the same: *be agile*. I strongly believe in learning from mistakes, iterating quickly, and building strength over time. I understand that big change is made from consistent small efforts, and strive to continuously improve myself every day.

When I work on things, I give them 100% of my effort. Regardless of whether I'm folding laundry or writing unit tests, I thoroughly enjoy myself and constantly push myself to new levels of understanding and efficiency.

Purpose

My main purpose is to make the world an awesome place. I want to solve cool problems, and help make Earth the best planet in the universe :)

To this end, my daily goal is to push myself out of my comfort zone a little bit every day, and continuously improve. I want to build and create software that *rocks*.

Personality

I think the most accurate way to describe myself is a "laid-back programmer":

- I enjoy wearing shorts, sandals, and t-shirts from ThinkGeek.

- I *hate* Windows.
- When I'm not programming I'm often attending programming events, or doing fun outdoor activities like CrossFit, biking, hiking, and exploring.
- I constantly build and contribute to open source projects.
- I read every day.
- I like writing on my blog.
- I've got an uncanny ability for finding relevant gif images during IRC conversations.

I'm friendly, fun, and nerdy.

Skills

I've been programming since I was just a young kid, playing Wolfenstein 3D. Over the years, I've worked with numerous technologies and learned various design patterns.

My greatest strength is my insane drive to learn new things and improve every day. Along with this, I've internalized the three virtues of a programmer, and constantly battle with myself to build software that satisfies my laziness, impatience, and hubris >:)

Python

Python is by far my strongest programming language. I've worked with it extensively for ~5 years, and am very familiar with the standard library, and open source ecosystem.

I've used Python to build:

- Small command line utilities.
- Professional software for large telephony companies.
- Numerous websites and web services.
- Open source libraries and tools.

I'm active in the Python community, frequently attend Python events, and work with other Python developers.

Django

I've been using Django (both professionally and personally) for the past 2 years. I've used it to build web applications for my company, as well as myself, and actively contribute to the Django ecosystem.

What I love about Django is its flexibility, best-practices driven design, and awesome documentation. I've personally adopted a lot of Django's development traits, and the framework has inspired me to become a better developer. Furthermore, the Django ecosystem is enormous, and it's really nice to use so many great applications written by other programmers.

Flask

I started using Flask when it was first released for small projects. To date, I've built 3 web applications with it (all for my company) that power small internal websites.

What I like about Flask is the simple project skeleton, and extremely in-depth documentation.

Celery

In the past year I've come to love celery, the distributed task queue. I've used it in my workplace to help scale our applications to support thousands of users and improve server response times, and I've used it personally to build beautifully scalable asynchronous applications.

I'm extremely familiar with the setup and usage of celery, and have used both celery and celerybeat to build high-performance applications.

puppet

After discovering puppet earlier this year, my life completely changed. I now use puppet for everything–personally and professionally.

I've used puppet to:

- Build and manage several large production, staging, and development environments.
- Manage complex software environments that support thousands of users.
- Automate sysadmin work and remove thousands of hours worth of manual labor creating, configuring, and managing servers.
- Automatically scale my work's cloud infrastructure to actively add and remove servers based on user demand.

monit

monit is an application monitoring and healing program that I've used to build fault-tolerant software stacks at work, and for fun. I'm familiar with the basic monit scripting language, and have written numerous scripts to monitor applications, ensure they're running properly (via numerous statistics such as load, memory usage, response time, etc.), and attempt to both auto-correct issues and send notifications when problems arise.

memcached

In order to help scale large web applications, I've made extensive use of memcached to improve application response time and scale computationally expensive software.

I've setup and maintained memcached server clusters, and have experience working with the python memcached libraries (specifically, with Django's caching backend) to rapidly scale python web applications.

Rackspace

As the lead developer at a tech startup, I've personally helped build a scalable cloud infrastructure using Rackspace Cloud Servers.

I'm familiar with their python APIs and CLI tools, and know how to easily automate server provisioning.

Asterisk

Over the past 3 years I've worked intimately with the popular open source PBX system, Asterisk.

In addition to designing, implementing, and maintaining complex telephony systems for various companies, I've also made several performance patches, bug fixes, and built helper libraries to make interacting with Asterisk easier for developers.

Heroku

If I'm not currently held captive by terrorists who insist I deploy code to their own servers, then I'll unquestionably be using Heroku to host my web projets.

To date, I've migrated multiple large sites from other cloud providers (like Rackspace) to Heroku, for the betterment of all society.

In all seriousness though: I love Heroku, and I'm an expert at running sites on it. I'm familiar with their deployment model, best practices, and most of their addons.