

# Resume/CV

## Devin Austin

devin.austin@gmail.com

970-290-6669

## Summary: I am a web software developer.

I build web applications and have experience with many common frameworks in Perl and Ruby.

I learn very well, and am extremely tenacious when it comes to getting things done and figuring things out. I've had a good deal of experience with high pressure situations that require fast action, precise movements, and critical thinking skills in order to get something done, or fixed. In my experience, good software is developed by combining research, planning, quick and iterative development cycles, testing and communication. I spend my time making sure I have the information required to build the best product that I can in the time given, and I make sure I have the tools and knowledge to do so.

In summation, you give me a specification, and I'm going to do everything in my power to produce a reusable, extensible, fast and efficient solution for you, on time, and with a smile.

## Skills

### Languages

- Perl
  - ◆ Web Frameworks/Technologies/Network
  - ◆ Catalyst
  - ◆ CGI::Application
  - ◆ Web::Simple
  - ◆ Mojolicious::Lite
  - ◆ Plack
  - ◆ Net::Twitter
  - ◆ Facebook
  - ◆ Database/ORMs
  - ◆ DBIx::Class
  - ◆ Search
  - ◆ Kinosearch
  - ◆ ElasticSearch
  - ◆ Lucy
  - ◆ Caching/Non-RDBMS Datastores
  - ◆ Cache::Memcached
  - ◆ Redis
  - ◆ Worker Queues
  - ◆ Resque
  - ◆ Gearman

- ◆ MooseX::Workers
  - ◆ OO Technologies
  - ◆ Moose
  - ◆ Moo
  - ◆ Class::Accessor::Fast
- Ruby
  - ◆ Web Frameworks/Technologies
  - ◆ Ruby on Rails
  - ◆ Rack
  - ◆ Unicorn
  - ◆ Sinatra
  - ◆ Database/ORMs
  - ◆ ActiveRecord
  - ◆ Sequel
  - ◆ Search
  - ◆ Tire (elasticsearch)
  - ◆ Worker Queues
  - ◆ Resque
- Javascript
  - ◆ Frameworks
  - ◆ YUI
  - ◆ JQuery

## Applications/Technologies

- Puppet
- Nagios
- Munin
- Varnish
- Memcache
- ElasticSearch
- Solr
- Resque

## Professional Experience

### January 2012 - April 2012 Ionzero, LLC

#### Software Engineer

Worked on a large, legacy Perl application that deals with airline flights locally and internationally.

Used homegrown ORM to build and execute queries on a MySQL database

Debug large portions of code without much documentation daily, addressing ways to implement new enhancements and maintain good programming practices.

## **September 2011 - December 2011 ExposureManager, LLC**

### **Senior Software Developer**

Worked on a CGI::Application+MySQL backed web application with a heavily jQuery oriented front end that dealt with large numbers (15k photo batches in some instances) of photographs. Worked on building a Catalyst application backed by PostgreSQL using DBIx::Class, Bread::Board, and Message::Stack.

Main highlights include architecting a sane git workflow for users who weren't familiar with git, as well as building a well rounded infrastructure for PostgreSQL database schemata, and the Catalyst web application. Also created a simple, but flexible JSON API that allowed for easy interfacing with Ajax requests and providing a scalable and easy to use internal application API, while remaining extensible and decoupled. Built a Facebook style "infinite scroller" for photo galleries using jQuery.

## **February 2011 - September 2011 SocialFlow, LLC**

### **Product Engineer**

Worked on a MySQL backed Catalyst+DBIx::Class application that dealt with enormous RDBMS based datasets (multiple gigabytes in any given instance).

Specifically, built a Catalyst application that maintained an up-to-date store of clients' twitter followers and calculated the percent change within a given time period. This made heavy use of backend parallel worker processes. I was able to get PostgreSQL to return results fast enough through DBIx::Class that we didn't have to enable any caching or other optimization solutions.

I also built a reporting tool for the company based upon jQuery's datatables plugin that acted as a company health and statistics source based on what users' activity was, financially and app based. This also involved a good bit of optimization (this time with a MySQL backend) in order to successfully retrieve a dataset for given users, that I was successfully able to accomplish. Heavy use of JSON and Ajax were involved in the creation of this tool.

## **May 2009 - February 2011 Ionzero, LLC**

### **IT Consultant**

Built an application to act as a proxy between a front facing php web application and a storage file system that handles upwards of 6,000 file uploads a day via the API that handles XML, JSON and plain HTTP data. Built using Catalyst and Plack. - wmcglobal.com

Managed a high traffic, enterprise level web application written in Ruby on Rails. This consisted of migrating that application to its current server, setting up a staging and production server from the ground up (SVN for version control and deployment management, Postgres database, nginx+mod\_rails for the Ruby on Rails application) - gohuman.com

Enabled database logging features through out a very large PHP codebase that tracked the number of reports being "passed" or "failed" after an initial audit. - wmcglobal.com

Set up MovableType pro and customized templates for a high traffic and respected technology blog. - catalyzed.org

Set up MojoMojo wiki running on FastCGI+varnish for caching for code snippets and examples. - catalyzed.org

## **January 2010 - April 2010 Nasa Pro Racing**

### **IT Consultant**

Managed and wrote a database and file data migration script in Perl to move data from a previously used OpenInteract application to a new Catalyst+DBIx::Class web application backed by MySQL and MogileFS.

Built a search engine feature that uses Data::SearchEngine and DBIx::Class to search based on racing class types, race tracks, and other such race track and car class based data.

Wrote a geocoding feature that calculates the distance of a track from a given city name or zip code and returns tracks within a chosen radius (5, 10, 15 miles etc) of city/zip code based on Geo::Coder and DBIx::Class

Implemented a materialized path structure for managing virtual directories for website assets and file data in the database based upon DBIx::Class::Tree::Ordered::MaterializedPath, allowing administrators to have a file system view of website assets.

## **April 2009-September 2009 Google Summer of Code**

### **Programmer**

Cleaned up a very large code base for the Catalyst::Helper API for the Catalyst framework. (Code can be viewed in the helperrefactor, moosifyapi and better\_scripts branches the Catalyst code repository [<http://dev.catalystframework.org/svnweb/Catalyst/>])

Moved all the Template::Toolkit data and image data out of the DATA portion of Catalyst::Helper and into a more sane File::ShareDir set up so that developers could create their helpers in a much easier fashion.

## **2008-Present Boyosplace.com**

### **Web Master**

Built a Catalyst based web application for a puppy photogallery site using DBIx::Class (ORM for use with multiple databases, MySQL in this case), Imager (for a custom photogallery/manipulation feature), Template::Toolkit, and a fairly comprehensive user management system to allow people to sign up, login, and post pictures of their puppies. Ran under fastcgi on nginx, and varnish for caching images. (now moved to github)

## **2006-Present codedright.net Loveland, CO**

## Senior Web Developer/Founder

Did consulting work for M3i Works that does web front ends for companies like Sun Microsystems

Have done multiple modifications to PHP driven websites including an image upload script for a t-shirt screen printing company

Submitted a patch to the MojoMojo wiki that allows users to switch between Textile and Markdown wiki markup.

Created a web crawler to index all electrical engineers from Australia's YellowPages using Web::Scraper. The time constraint was great on this, less than 36 hours, and the client was very happy with the result.

## Education

### 2007-2010 Metropolitan State College of Denver Denver, CO

BS in Computer Security with a minor in Computer Science

Majored in Computer Security with a Computer Science minor Bachelor of Science in Computer Security and Crime

President of local ACM chapter at MSCD.

Participated in CANVAS - A network security exploitation convention/workshop two years in a row. Contributed greatly to breaking into servers via multiple exploits, SQL injection, DCOM holes, etc. Placed 4th, and 3rd respectively.

### 2006-2007 University of Northern Colorado Greeley, CO

Majored in Business Administration in Computer Information Systems Participated in Lead On, a leadership camp that accepted only 100 incoming freshman. Activities included developing leadership skills, social skills, and preparing yourself for the adversities of college.

## Open Source Work

### Author of:

#### Perl modules:

- Form::Sensible::Reflector::DBIC
- Catalyst::Model::ElasticSearch
- [Grimlock](#)

### Maintainer of:

## Perl modules:

- Catalyst::View::Email (co-maintainer, but have been doing a lot of the maintenance work as of late),
- Catalyst::Plugin::Data::FormValidator,
- DBIx::Class::Ordered::Tree::MaterializedPath (pending release),
- and many more

## Software I've written:

- [Grimlock](#) - A simple, but extensible CMS written using Catalyst, DBIx::Class, Template::Toolkit and YUI3 for user interface and AJAX functionality.
  - ◆ Uses DBIx::Class::MaterializedPath for threaded comments
  - ◆ Well tested
  - ◆ Easy to use JSON API
  - ◆ YUI3 provides useful, fast, and quickly modifiable datatable functionality
  - ◆ Auto-save draft functionality
- Deimos - <https://github.com/dhoss/Deimos> A Catalyst+DBIx::Class based CMS that does all the things that I haven't been able to find in other CMS software (wordpress, MovableType), such as:
  - ◆ runs smoothly under nginx
  - ◆ provides a reasonable and intuitive Perl API, as well as a web application API (emphasis on JSON)
  - ◆ works nicely with PostgreSQL
  - ◆ ability to switch worker process and search backends easily (configurable stores)
- \*[PerlFu](#) A Catalyst based forum application that uses materialized paths to store threaded discussions.
  - ◆ quick storage and retrieval of threads and posts
  - ◆ XSS security precautions
  - ◆ Data::Manager for data validation and message passing
  - ◆ DBIx::Class::MaterializedPath for materialized path handling
  - ◆ Plack for easy deployment.
  - ◆ branch for parsing large XML datasets using XML::CompactTree::XS and XML::LibXML::Reader, as well as XML::Toolkit for web app interaction with smaller XML datasets.

Involved in the development of the Catalyst web application framework, DBIx::Class ORM, Moose OO framework test contributor

## Identities and resources

perlmonks.org nick: dhoss, same on IRC.

CPAN: DHOSS

Github: <http://github.com/dhoss>

Writing

Published many articles regarding perl, a few of which are listed here:

[Simple Photo Gallery in Catalyst](#)

[Next Page Redirect after Login](#)

[Plack and Catalyst](#)

[Trees in SQL with Catalyst](#)

[The New Catalyst ScriptRunner API](#)

## **Interests**

Hockey, running, football, weight lifting, computers, and target shooting, Kung Fu (student instructor for Pai Lum)