PATRICK KING

Full-stack web developer in Ruby on Rails + React + Node.js

patrick.f.king@gmail.com | *Calgary, Canada*
(403) 922-4960 | *This Resume: patrickking.github.io*

EXPERIENCE

Hacquity - Hacking Gender Equity Hackathon - May to Sept 2018

DoM Citizen (Github) 2018

Ruby On Rails | Postgres SQL | S3 | Google Docs API | Email (Sendgrid + Mailgun) | Heroku

Role: Lead Developer - **Challenges:** As part of an extended hackathon I built DoM (Department of Medicine) Citizen, a prototype community portal for physicians in Calgary. This rails app has features for sharing and reviewing trusted service providers, uploading and searching resumes, and connecting with potential mentors. Trying my hand at some new cloud APIs was interesting, and as the webapp is part of a larger initiative to study barriers to gender equity in medicine, I had the opportunity to help facilitate a workshop about the site and present it to potential users there. Working alone to deliver some intricate features, I opted to limit complexity and simulate a single-page app experience with turbolinks. I built a fast and responsive site with virtually no javascript.

VizworX Inc. - March 2013 to Present

Imports & Exports of Energy Products to and from Canada NEB (Github) - 2017 to 2018

React.js | Immutable.js | Redux | Webpack | IIS

Role: Consulting Developer - **Challenges:** While transitioning out of the company, I bootstrapped this new visualization and assisted with scoping and estimation before handing it off to the team. The main challenge was to offload and document all of the knowledge I had built up of the client and our projects. Keeping up with code review also became interesting, as I was no longer the author of any of code, and had less frequent contact with the team over time.

Pipeline Incidents (NEB) (Github) 2017

React.js | Immutable.js | Redux | Webpack | IIS | Node.js

Role: Lead Developer – **Challenges:** I onboarded a new junior developer, while transitioning the team to a new React based stack, while delivering a complex browser data visualization app under a tight deadline. My organizational skills, time management, client interfacing and detail orientation were tested on this project! We delivered the app successfully and on time, with a limited amount of overtime. I lead the adoption of code review as a practice at Vizworx during this project.

Exploring Canada's Energy Future (NEB) (Github) – 2016 to 2017

Coffeescript Browserify D3.js IIS Node.js Phantom.js

Role: Lead Developer – **Challenges:** I stepped into this project after it was already underway. As a pilot project for this client we had wide latitude in our choices of technology and approach, but we had to deliver this browser based visualization under tight deadlines and with limited access to the client's hosting environment. Challenges included building the app with tools not well suited to browser app development (D3 especially), integrating Node.js applications into an IIS hosting environment, and interfacing with an external design team and external client staff. With our partners focused on the design of the visualization, ensuring that non-design related concerns were considered and scoped was also a challenge.

Ivrnet Central – 2016

Ruby On Rails RSpec Docker Postgres SQL

Role: Developer – **Challenges:** This was a short engagement with another Calgary software company to support their development team; the main challenge was to drop into an unfamiliar Rails codebase and make worthwhile contributions during the six week span, while also learning the basics of Docker.

Geoviz – 2015 to 2016

React.js Flux Leaflet Node.js Objective-C + iOS C# + .NET ArcGIS (Server + SDKs)

Role: Developer – **Challenges:** Geoviz was a sprawling prototype project, with applications all communicating with each other on Windows hosts, iOS devices, servers, and in the browser. The essential challenge was keeping all of these platforms in sync as new features were added and communication protocols changed. Another notable challenge was implementing a React based mapping application at a time when adapters for Leaflet and Mapbox were at an early stage of development, or did not exist at all. Adapting DOM-state heavy and procedural tools like Leaflet to React's less stateful rendering approach was a challenge.

WeEmploy – 2014

- Ruby On Rails
- Postgres SQL
- RSpec
- Sunspot Solr + Lucene

> **Role:** Developer – **Challenges:** The application concerned an online job board, and one notable challenge was designing a search system integrating Rails and Solr for matching candidates to jobs based on skills and other traits. The client brought their own design team with some ambitious ideas, which demanded solid client facing abilities and presented some interesting HTML/CSS implementation challenges.

Jobbsite – 2013 to 2015

- Ruby On Rails
- Postgres SQL
- RSpec
- Cucumber
- Phantom.js
- Selenium
- Paperclip
- Prawn
- Ampersand.js

> **Role:** Developer – **Challenges:** I began on Jobbsite as a junior developer working under a talented senior to build a custom time and order management system for a client, and ended as a leading developer on a larger team adapting the software to serve as a SAAS product. There were numerous challenges over the years, including onboarding additional junior devs, adding a rich client experience using a browser toolkit called Ampersand.js, the Rails 3 to 4 upgrade, a continuous need to optimize the speed of a sprawling test suite, balancing the needs of the client vs. the needs of the product, and implementing a reliable multi-tenant database system.

Agile Software Engineering Lab, University of Calgary (Dr. Frank Maurer) – May 2011 to February 2013

NSERC Surfnet Research Network Page – 2013

- Ruby On Rails
- Postgres SQL
- Paperclip

> **Role:** Lead Developer – **Challenges:** My last task with Dr. Maurer's lab was to implement a website to document the work of the Surfnet research network, where network members could log in and upload descriptions and papers detailing their projects. This was my first Ruby on Rails project. Effectively, I attempted to learn Rails and implement a simple CMS on my own in two months, and in retrospect this goal was entirely too ambitious. Not knowing this at the time, I made a serious attempt to complete it. I learned a lot while building a substantial portion of the site, before handing it off to another intern.

MRI Table Kinect – 2012

- C# + WPF
- Kinect
- Objective-C + iOS

Role: Developer - **Challenges:** Building off of MSE-API and working for a grad student in Dr. Maurer's lab, I built this prototype which synchronizes the display on a touch tabletop with an iPad by tracking the device with a Kinect. The main challenge was the short timeframe, the project came together in one week. The work resulted in a publication in MediCHI'13.

Multi-Surface Environment API (Github) - 2012

C# + WPF Node.js Kinect Objective-C + iOS

Role: Developer - **Challenges:** As one developer in a team of six, we built a framework for cross device communication incorporating Kinect skeletal recognition and gestures on mobile devices. The main challenges included keeping codebases for different platforms in sync, and coordinating work between a medium size team of students and interns who had many other demands on their time. In and around work on this project, I started a drive within the lab to move our source control from an in-house TFS Version Control server to Github.

eGrid - 2011

C# + WPF ArcGIS (C# SDK)

Role: Developer - **Challenges:** eGrid was a prototype power grid management application, for tracking work crews and power equipment issues / outages citywide. The main challenge was diving into an existing codebase while learning C#, touch interface APIs, and GIS basics. eGrid ran on an early generation touch tabletop with early Microsoft touch software, both of which came with unique issues to work around.

University of Calgary iGEM Team - May to August 2009

Synthetic Biology Interactive (Second Life Installation)

LSL (Second Life)

Role: Team Leader - **Challenges:** I lead a team of three other students to build a virtual installation about synthetic biology in Second life, including two biomedical sciences students with no software development experience, mentoring and guiding these students in the basics of coding and 3D graphics was a key challenge. An interesting technical problem was working within the limited environment provided by Second Life's integrated Linden Scripting Language: each script is limited to 64kiB for compiled code and stack / heap memory. But, as a garbage collected language LSL includes no way to manage memory directly. Adding a line of code to a script, or adding one too many elements to a list, could result in an out of memory error. I split functionality across multiple scripts to access more memory, and this required the addition of an IPC-like communication system between scripts over the game's chat channels.

EDUCATION

BHSc - *Bioinformatics (Honours) - University of Calgary - 2012*

Honours Thesis: *An Algorithm for Chromatin Immunoprecipitation Sequencing Analysis.*

Python R

SKILLS + TECH

What I'm Best At

React.js Node.js Ruby On Rails RSpec Cucumber Javascript (ES 2015+) ESLint Coffeescript Immutable.js Redux + Mobx + Flux Git (CLI + Git Flow) Webpack D3.js Agile Practices Promises + Bluebird.js jQuery

What I'm Good At

Postgres SQL IIS Phantom.js Python C# + .NET (Web and Windows desktop) WebGL + Three.js Team Foundation Version Control (TFS) Leaflet Browserify Heroku ArcGIS (APIs + SDKs + Server Configuration) Prawn French (Written + Spoken)

What I've Played With

S3 Google Docs API Email (Sendgrid + Mailgun) Java + Android Objective-C + iOS Docker Sunspot Solr + Lucene R MongoDB Go Ember.js MSBuild Ampersand.js GLSL (Witih WebGL) Puppet Meteor.js LSL (Second Life) Drupal SQL Server (TSQL) C + C++ Django Kinect Blender Selenium

OTHER WORKS

Accessible Visualizations: A Case Study (slides) - I presented work on making the Energy Futures visualization accessible, as part of the our design partner's Data Empowerment Speaker Series.

"Star Trek: Armada" Gallery - A shrine for an old computer game I enjoyed.

Coffeescript Node.js WebGL + Three.js

Neoderelict – A simple prototype browser game.

Coffeescript WebGL + Three.js Blender

References available on request