Sean Moran-Richards

Senior and Lead Developer–full stack Rails and React

Bath, UK
+44 78 6977 0295
sean@flyinggrizzly.net

EXPERIENCE

Lead Developer - Mystery Applicant, Bath UK

September 2021 - PRESENT

Took over lead of product development and dev team from my mentor. I balance continuing to deliver work personally, while coaching and mentoring two direct reports.

Developer - Mystery Applicant, Bath UK

June 2018 - September 2021

Developing and maintaining a Rails + React application that performs hundreds of thousands of surveys annually. I lead work to replace an aging visualization dashboard with a TypeScript+React+D3 frontend that is faster for users, more extensible, and easier to maintain. I also helped replace aging ETL processes with more streamlined and reliable AWS Lambda workflows.

Junior Developer - University of Bath, Bath UK

November 2017 - June 2018

I worked on the University's two custom content management systems, and updated the CI and deployment workflows in cooperation with developers from other teams across the University.

Digital Supporter - University of Bath, Bath UK

September 2016 - November 2017

Providing first-line support to internal and external web users at the University; assisting the development team with agile workflow and smaller Ruby development tasks. Prototyping tools in Rails.

Business Specialist and Expert - Apple, London UK

September 2015 - September 2016

Develop close relationships with new and existing customers; support B2B customers with MDM rollout preparation. Provide training and support to team in store; drive sales on frontline with both consumer and B2B customers.

SKILLS

Delivering and improving maintainable, readable code

Mentoring and peer review

Test driven development

Agile development practices and principles

Ruby, Ruby on Rails, RSpec

JavaScript, TypeScript, React, D3.js, Redux, Jest, Enzyme

MySQL, Postgres

AWS Elastic Beanstalk, RDS, S3, Route53, CodePipeline and CodeBuild, Cloudfront, Lambda and Step Functions, Serverless Applications, Cloudformation

Docker, Docker Compose

LANGUAGES

English - native speaker

Spanish - conversationally fluent

Japanese - learning

LEGAL AND ELIGIBILITY

UK Citizen, US Citizen

EDUCATION

London International School of Performing Arts (LISPA), London— *Certificate in Creating Theatre and Performance*

September 2010 - June 2012

Post-graduate program focusing on physical and masked performance, and collaborative devising. All work was written by students during rehearsal processes collaboratively.

Wesleyan University, Connecticut — *B.A in History of Religion*

September 2006 - June 2010

Social science degree with focus on subjective human experience of community through religion. Final GPA of ~98%; rough equivalent of UK high Upper Second (conversion source: Fullbright Commission).

LOMINGER COMPETENCIES

Attention to detail

Organizational Agility

Intellectual horsepower

Self development

Interpersonal savvy

Composure

Technical learning

PROJECTS

Rewrite of Mystery Applicant client frontend with React and D3

When I joined Mystery Applicant, the client frontend was an old Rails view and controller action that could take nearly a minute to load because of the complex and convoluted data calculation that happened prior to render. It was difficult to change, and doing so could also lead to easily-missed regressions or even incorrect data.

The rewrite project consisted of learning React and D3 (and later TypeScript), and rewriting the dashboard to be modular and fast. The single complex and sprawling data calculation in the controller was replaced by multiple small React container components that request only the data required data from our backend. By XHR The components then pass the data down to a D3 component.

React and D3 do not work together cleanly out of the box, but we found a solution that uses React's ease of layout and optimized updates, and still allows for idiomatic D3 code for visualization. This means we get the speed of working with React, and retain access to the numerous tested and reliable charts D3 is capable of creating.

In addition to the visualization work, this project also required the creation and evolution of a Rails API and TypeScript request interface that provided a standard and clear way for all the React containers to fetch data. A strong focus was put on keeping them similar to each other to reduce the cognitive load when moving from the frontend to the backend (and vice versa), while also respecting the differences in convention and pattern inherent to each language so that other developers would not be caught out with unusual JS or Ruby code.

Update of survey interface to improve applicant experience

Using the lessons learned from the dashboard project (above), I upgraded our survey interface using React and XHR requests to make it easier for respondents to use. Previously, they would be presented with a long list of questions in a form all at once, which submitted a single response to the Rails backend.

Updating this to show one question at a time has two benefits for us. First, respondents are not shown too much information at once, potentially improving the quality of responses as cognitive load lessens. Second, showing a

single question at a time has allowed us to extend and improve our question logic, including opening up the possibility of conditional questions or branching question paths that respond to respondents' prior answers.