Sean Moran-Richards

Staff/Lead Developer–full stack Rails and React

Bath, UK
+44 78 6977 0295
sean@flyinggrizzly.net
github.com/flyinggrizzly
UK Citizen, US Citizen

SELECTED EXPERIENCE

Staff Developer - Shopify, Bath UK/Remote

February 2025 - PRESENT

Delivering new product features based on improved architecture, ensuring forwards compatibility of new features. Mentoring midweight and senior developers. Identifying missing features and new platform and product work to build the platform for the long term.

Senior Developer - Shopify, Bath UK/Remote

February 2022 - February 2025

Maintaining legacy features used by tens of thousands of merchants while rebuilding underlying architecture to improve scalability and external feature integration. Mentoring midweight and junior developers. Identifying platform bugs and issues and making prioritization decisions about them against existing work.

Lead Developer - Mystery Applicant, Bath UK

September 2021 - December 2021

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. Company entered administration in December 2021.

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.

SKILLS

Delivering and improving maintainable, readable code

Mentoring and peer review

Test driven development

Agile development practices and principles

LLM/AI aided development

Ruby, Ruby on Rails, RSpec, Minitest, GraphQL

JavaScript, TypeScript, React, D3.js, Jest

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

PROJECTS

Enabling price overrides, improved completion errors and corrections in Shopify Draft Orders

I lead work to further improve the capabilities of Draft Orders after replatforming onto Checkout Extensibility.

Previously a draft order would “freeze” merchandise pricing, even if the actual product had been updated in the meantime. For some users this was a feature, and others it was an anti-feature.

We did careful work to make the necessary changes for this to be an explicit override in the checkout process, make it clear to merchants when prices are frozen or floating, and added guardrails to avoid disrupting merchant-customer relationships when we rolled out the change (a lot of work was put into prevent putting the merchant-customer relationship at risk in the case prices might unexpected start floating).

In addition, we built out a framework to translate the usually buyer-specific error messaging of Checkout into merchant-relevant messages, and improved the workflow merchants experience when some stale state prevents a checkout. This includes offering to help them freeze in the old price, or update the draft to use the new price.

Adding discount code support to Shopify Draft Orders

Following on from replatforming Draft Orders onto Checkout Extensibility, I lead the project to implement one of the first new features based on it: discount codes.

I made sure that UX and Product understood the new constraints that the Checkout Extensibility platform placed on our product, in order to guide the product and design process to avoid working against the new engine.

I also made sure that our domain implementation adhered closely to the contract of Extensibility, so that any future changes there could be safely accommodated. This has already paid off, since more recent updates to the discount data modelling have required only a few days of work to incorporate, delivering new features rapidly to merchants, instead of the months-to-years delays they would have seen previously in Draft Orders.

Replatform Shopify’s Draft Order engine to Checkout Extensibility

Shopify’s Draft Orders used to be an isolated system that was only capable of pushing order and checkout information into the rest of the platform, and any time we wanted to add a feature (like discount codes), it had to be rebuilt from scratch.

In order to maintain consistency with the rest of Shopify, we replatformed Draft Orders onto the Checkout Extensibility engine to consume all data directly from the canonical truth–the final order creation process. The goal being that any and all decisions we made mapped as closely as possible to Storefront checkouts, so we could ensure ongoing consistency and reduce the work required to add new features.

All of this was done without interruption to users through extensive testing, and frequent smoke tests.

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 made two improvements. 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 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.

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).