

Matt Valley

Full-stack Software Engineer, Tech Lead, Indie Hacker
hi@mv.id.au | 0411171487 | [Github](#) | [LinkedIn](#)

I hold 20+ years of experience in software development. Throughout my career, I have leveraged many programming languages & technologies and played many roles. Today, Javascript & Golang are the languages I use for commercial projects.

For front-end development, I've used modern Javascript, Typescript, React (Vanilla React, Next.js, Remix), and CSS (Vanilla, BEM, CSS-in-JS, Tailwind) to build design systems, Single-page applications, web apps, and Javascript modules.

On the back-end side, I have built a range of GraphQL, Restful, CLI, Task Workers, Real-time socket-based services, and other back-end systems using Node.js (Vanilla, Nest, Express etc.), and Golang, leveraging Docker & AWS Lambda as the host.

DevOps-wise, I've extensively leveraged AWS, including ECS, AWS Lambda, CloudFront, S3, SES, SNS, EventBridge, CloudWatch, Pinpoint, Kinesis, DynamoDB, RDS, KMS, CodeArtifact, SQS, CloudFormation to provision, deploy & run front-end and back-end applications, primarily using Terraform as my IaC solution.

I've extensively used RDBMS, NoSQL and Document databases, including SQL Server, Postgres, MySQL, MongoDB, CouchDB, and DynamoDB in large-scale production systems.

Test-driven Development, Clean architecture, scalability, readable code and maintainable software are core concepts I practice, advocate and appreciate as a software engineer.

Work Experience

- Founder, Testfully.io
- Senior Software Engineer, THE ICONIC
- Senior Tech Lead at Westpac
- Engineering Lead at Different
- Senior Software Engineer at Autopilot
- Senior Software Engineer at Tabcorp
- [+ 4 more positions](#)

Technical Talks

- [Reactive Programming using Rxjs in React](#)
React Sydney meetup, May 2021
Brisbane Javascript meetup, June 2021
- [GraphQL at Different](#)
SydJS meetup, January 2020
GraphQL Sydney meetup, December 2019
- [Metaprogramming in Javascript](#)
Web Directions Summit, November 2017
- [Composition in React](#)
Canva meetup, October 2017
- [TDD: Develop, Refactor and Release with confidence](#)
React Sydney meetup, February 2017
- [Modular development in Node.js](#)
Node Ninjas meetup, July 2015
- [Database Replication in MongoDB](#)
MongoDB Sydney, May 2015
- [CoffeeScript, An Introduction for Node.js Developers](#)
Node Ninjas, Oct 2014

Open Source

- [variant-classnames](#)
- [Strummer-plus](#)

Recent Work Experiences

Testfully.io

December 2020 / Present / Indie Hacker

Testfully is an API Client / Synthetic API Testing & monitoring platform. It directly competes with tools like Postman, Insomnia and Hoppscotch. Testfully is a labour of love, a company I built from the ground up and nurtured daily. Below are some of my development work for this venture so far:

Design System

A unique design system for Testfully's SPA apps.

- Developed using React, Typescript, Storybook, Radix UI, and Tailwind with dark mode support
- Leveraged Nx workspaces for modularity of codebase
- Packages deployed to AWS CodeArtifacts via Github Action pipelines for ease of use by others
- Practices Composition Pattern for reusability

Customer app

Testfully's main app; available via app.testfully.io.

- Developed using React, Typescript, Akita, RxJS, React Router, above Design System, and Tailwind
- Leverages Yarn Workspaces for modularity
- Deployed via AWS CloudFront using Github Action
- Unit tested heavily using Jest & Chai.
- The Desktop version uses Tauri (Electron alt), available for Windows, Mac, and Linux.

The back-end for Front-end

A back-end exclusive for app.testfully.io

- Written in Golang, available via HTTP & Web Socket
- Communicates using messages (similar to tRPC)
- Deployed via AWS ECS Fargate, ALB, and WAF, leveraging Terraform for Infrastructure provisioning

Task Processing System

Handling background job for Testfully

- Written in Golang & Node.js (Serverless Framework)
- Powered by AWS EventBridge and SQS.
- Deployed using ECS Fargate and AWS Lambda. Provisioned using Terraform.

GraphQL API

Serving both customers and Testfully apps

- Written in Golang & Node.js, with a federated schema using Apollo.
- Deployed using AWS ECS Fargate, ALB, and WAF.
- Provisioned using Terraform.
- Redis, Postgres, and DynamoDB are the primary DBs.

Testfully CLI

Brings Testfully to CI/CD pipeline of customers

- Written in Golang for portability and performance
- Uses Github Action Pipelines for producing binaries
- Deployed to S3; provisioned using Terraform

Testfully Browser Extensions

By-passes browser limits via native vendor API

- Written in vanilla JS, leveraging vendor-specific API
- Deployed to Chrome, Edge, and Firefox stores

Infrastructure As Code

Provision and maintenance of DEV, UAT, and Prod versions of the platform on AWS

- Leverages Terraform
- A CLI was developed using Bash to make it easier to work with Terraform templates.

THE ICONIC

July 2021/ September 2022 /Senior Software Engineer

I was hired to help the company execute the plan for rewriting theiconic.com.au in React.

Design System

Bringing consistency across the board.

- Developed using React, Typescript, Storybook, and Tailwind – heavily relying on React Hooks.
- Practices Composition Pattern for reusability
- Leveraged Nx workspaces for modularity of codebase

Micro front-end Boilerplate

A boilerplate powered by Nx to enable cross-functional teams to build micro front-end apps.

- Developed using React, Typescript, Nx, and Next.js
- Comes with State Management (Akita), Feature Flag (LaunchDarkly), Logs, Authorization (JWT), Error Handling, Routes
- Deployed to ECS Kubernetes, uses Terraform.
- Shipped Store Credit Page and Wishlist apps to production, leveraging the same boilerplate

GraphQL API

Serves micro-frontends, proxies backend services

- Developed using Nest, following a microservice architecture made possible using Federated GraphQL schema (Apollo)
- Deployed to ECS Kubernetes, leveraging Terraform
- Maintained on top of an Nx workspace for modularity
- Comes with a custom boilerplate to ease the process of creating new microservices.

Westpac

February 2020 / July 2021 / Senior Technical Lead

I was hired to execute the company's Smart Desktop plan, a group of desktop apps powered by React and OpenFin.

Design System

Bringing consistency across the board and accelerating the development of new apps.

- Developed using React, Typescript, Storybook, and Tailwind – heavily relying on React Hooks.
- Practices Composition Pattern for reusability
- Leveraged Yarn Workspaces
- Deployed to the company's internal code registry

Request For Quote app

A real-time app that connects sales & traders within the WIB division of Westpac.

- Developed using React, Typescript, RxJS, and Tailwind
- Interacts with the backend via WebSocket
- Unit tested extensively using Jest and Chai

Openfin & FDC3 Adoption

Adopted the Openfin platform and implemented required features using this platform. FDC3 is a popular data exchange specification in the banking software industry, allowing cross-app communication between vendors and internal desktop apps. I have successfully incorporated this spec into the implemented apps.

Different

September 2018/ May 2020 / Technical Lead

Joined the startup during its early days to establish an offshore & onshore engineering team and take over the tech stack from an offshore software consultancy.

GraphQL API

Served the company's mobile and web apps for tenants, owners, contractors, and internal staff.

- Written in Javascript using GraphQL.js and Flow for strong typing / MongoDB & MySQL for database
- Unit & Integrated tested using Mocha & Should
- Deployed using AWS ECS and ALB, leveraging Terraform for provisioning
- Jenkin pipeline for CI/CD

Design System

Making the development of features faster and better maintenance over time.

- Developed using Typescript, React, BEM, and Saas
- Unit & Integrated tested using Jest
- Published as private NPM packages via Jenkins pipelines

Task Processing System

Handling background jobs and report generation

- Written in Typescript using Node.js
- Unit & Integrated tested using Jest
- Powered by SQS, Deployed via ECS

Infrastructure + CI/CD

Provision of 4 identical environments in AWS using Terraform. Reusable Jenkins-based pipeline (Jenkinsfile) on top of shell scripts.

Statement Engine

Produces financial statements for property owners

- Written in vanilla Typescript; extensively tested due to the sensitivity of the engine
- Supports all major Property Management scenarios out of the box (Management Fee, Maintenance, Advertisement Fee, Legal Fee, Rental Income, GST)
- Generates and stores statements every month for managed properties
- Produces PDF files on demand using AWS Lambda

Ledger Engine

Produces rental ledger for a lease

- Written in vanilla Typescript; extensively tested due to the sensitivity of the engine
- Supports all major Property Management Rental scenarios of the box (Bond, Rent Split, Over Payment, Under Payment, Maintenance)
- Offers Restful API for other backends

Payment Engine

A service that collects rents

- Written in vanilla Typescript; extensively tested due to the sensitivity of the engine
- Supports multiple payment providers out of the box – making the switch from one provider to another easy
- Multiple safeguards to prevent fraud, and double charging tenants

Established the Engineering Team

- Established an engineering team in Sri Lanka and hired 15 full-time engineers.
- Trained engineers to the desired standards of the company
- Have done regular 1-1 with engineers and managed promotions, bonuses, and salary reviews

Autopilot

February 2016 / July 2018 / Senior Software Engineer

I was hired as a full-stack engineer to work on the company's new dashboard.

Design System

Autopilot's UI is rather complex to build; the goal was to make it as reusable as possible using a UI lib.

- Written in Javascript, React, Saas, and BEM
- Unit and Regression tested
- Supported mobile variation

Dashboard V3

Replacement of the existing dashboard with a brand-new dashboard to support the company's new features.

- Written in Javascript, React, Saas, and BEM
- Unit and Regression tested
- Leveraged Redux and React Router
- Offers Real-time capabilities using WebSocket
- Deployed using AWS Cloudfront and S3, using Terraform for provisioning resources

Backend for Front-end

A web-socket based backend for the dashboard

- Written in Javascript using Node
- Offers real-time sync of multiple users working on the same dataset
- Deployed to AWS using EC2 instances

Visual Chart Service

Generates JPG charts on the fly for a weekly Digest newsletter that customers would receive.

- Written in Javascript using Node
- Powered by AWS Lambda & Serverless Framework

Tabcorp

Feb 2015 / Feb 2016 / Senior Software Engineer

I was part of a team that migrated Tab's digital capabilities from legacy Java/Flash to AngularJS & Node.

Restful API

Development of various wagering, payment, and account API endpoints

- Developed API endpoints in Coffeescript & Node
- Unit & Integrated tested due to the nature of business
- CloudFormation was used for the provision of services
- Deployed to EC2 instances

Quick Multi Dashboard

A front-end dashboard for bookies to configure multi bets for the website

- Written in Javascript, Angular, and Bootstrap

Onit Media - Lead Software Engineer

June 2013, September 2014 / Senior Software Engineer

Developed the Shuuga app, a geo-based app that finds local deals (happy hour, discount) for users

Shuuga API

- Developed using Node.js
- Deployed to EC2 instances

Shuuga Dashboard

Admin dashboard for the internal staff

- Written in AngularJS & Bootstrap
- Deployed via AWS S3

