



Vladimir Logachev

Location: Armenia (remote)

Mail: vladimir@logachev.dev

GitHub: <https://github.com/vladimirlogachev>

🔧 Creating products. 🍷 Engineering every day.

Experience

Data Engineer

Samokat.tech

<https://samokat.tech/>

09/2023 — present

Samokat.tech develops solutions for real-time retail (the Samokat food delivery app and other marketplaces).

I develop data pipelines using platform instruments.

Responsibilities: engineering, code review

Backend: Scala, Spark, Kafka, MinIO, Airflow, MS SQL, Jenkins

Founder and Engineering Lead

Wolf

08/2021 — present

The project is a full-stack web application for trading on the Binance crypto exchange. It consists of a trend-following trading algorithm, exchange simulation engine for backtesting, exchange API client for trading with real account, and a web application for inspecting and debugging the trading algorithm, and displaying stats.

We use several kinds of testing techniques, including snapshot testing and property-based testing, along with case-based testing. We have schema-first typesafe SQL queries to PostgreSQL. And the haskell-to-elm library generates Elm code from Haskell code. Also, we have a strong focus on the absence of partial functions. This combination of technologies allows us to catch most regressions in the compile-time.

The line count is ~17k of Haskell and ~10k of Elm code.

During a technical interview, I can demonstrate the project running in a production environment and its source code.

Responsibilities: design, technology strategy, project management, leading, mentoring, engineering, testing, code review, devops

Backend: Haskell, servant-server, postgresql-typed, haskell-to-elm, mtl, lens, conduit, relude, hspect, QuickCheck

Frontend: Elm, elm-ui, elm-charts, remotedata, elm-review, elm-test

Infrastructure: Kubernetes, Docker, Nginx, Grafana, Loki, DigitalOcean, GitHub Actions, GitHub Container Registry

Backend Team Lead

Swift Invention

<https://www.swiftinvention.com>

08/2022 — 05/2023

I maintained two projects as a team lead, including implementing new features, writing tests, and fixing bugs.

Also, I mentored new developers to get on board with Scala (mostly, from a Java background), and ran a weekly Scala book reading club.

Responsibilities: leading, mentoring, technical interviews, engineering, testing, code review

Backend: Scala, zio, zio-http, zio-test, tapir, circe, chimney, enumeratum, flyway, testcontainers, finagle, scalatest, MySQL, Redis, Docker

Software Engineer

Fourier Labs

<https://fourierlabs.io>

03/2022 — 04/2022

I worked on the POC EVM implementation for the Cardano blockchain.

Responsibilities: engineering, testing, code review

Backend: Haskell, Plutus

Infrastructure: Nix

Blockchain: Cardano, Ethereum

Frontend Engineer

Pamir

05/2020 — 12/2020

Developed a web application, which utilizes server-side rendering and covered it with unit tests. Packaged everything in Docker and set up CI.

I also mentored the second frontend developer who joined the team later.

Responsibilities: mentoring, engineering, testing, code review

Frontend: TypeScript, React, Next.js, GraphQL, Apollo, FP-TS, Emotion, Jest

Infrastructure: Nginx, Docker, GitHub Actions

Software Engineer

Eldis

<https://eldis.ru>

10/2019 — 12/2019

I developed a declarative decoder for the internal binary document format, covered it with tests, including property-based testing.

Responsibilities: engineering, testing

Backend: Scala, scodec, cats, fs2, decline, specs2, scalacheck

Fullstack Engineer

Neolab-Nsk

01/2019 — 09/2019

I implemented new functionality in existing web applications, fixed defects, and developed new applications, and microservices, covering them with unit tests and integration tests.

Responsibilities: engineering, testing

Frontend: TypeScript, React, Redux, Saga, RxJS, FP-TS

Backend: TypeScript, Node, Redux, Saga, RxJS, Redis, Lua, Mongo, PostgreSQL, Clickhouse, Docker

Frontend Engineer

SocialSweet Inc

<https://sweet.io>

08/2018 — 01/2019

Sweet's product is a loyalty platform, social network, and online store.

I performed tasks related to business logic at the front end and was engaged in covering the existing code with unit tests and tuning them, thanks to which the tests were launched using CI pipeline, and the defects associated with an unsuccessful merging of Git branches in a huge codebase really began to be prevented.

Responsibilities: engineering, testing, code review

Frontend: TypeScript, Angular, RxJS

Frontend Engineer

Allmax

<https://savl.com>

11/2017 — 08/2018

I worked in the Savl project — this is a mobile application, a wallet with support for 6 cryptocurrencies.

I was responsible for the data layer in the mobile application. I applied everything that I learned from books about functional programming and software design, and also completely covered the business logic with tests, as a result of which the application became fault-tolerant and modular, that is, it stopped crashing due to exceptions or unexpected behavior of external services, and allowed to enable and disable support for individual cryptocurrencies at any time.

Also inside the company, I made several presentations on functional programming.

Responsibilities: mentoring, engineering, testing, code review

Frontend: JavaScript, Flow, React Native, Redux, Saga, Ramda, Sanctuary, Socket.io

Open Source Contributions

haskell-to-elm/servant-to-elm-example

<https://github.com/haskell-to-elm/servant-to-elm-example>

An example full-stack web app, built in a code-first, typesafe, and functional way. The servant-to-elm generates types and decoders/encoders from Haskell types and Servant definition to Elm, which not only catches regressions in the compile-time but also provides ready (and highly configurable) Elm functions to fetch necessary data from the server.

Haskell, Elm, servant-server, haskell-to-elm, servant-to-elm

higherkindness/mu-graphql-example-elm

<https://github.com/higherkindness/mu-graphql-example-elm>

An example full-stack web app, built in a schema-first, typesafe, and functional way. The mu-haskell library typechecks Haskell code against GraphQL schema at compile-time. The elm-graphql library generates types and decoders/encoders for Elm from GraphQL schema. I rebuilt its Elm frontend and made minor changes to the Haskell backend (and also discovered a couple of bugs in the mu-haskell library itself).

Haskell, Elm, GraphQL, mu-haskell, elm-graphql

Russian translation of the Mostly Adequate Guide to Functional Programming in JavaScript

<https://github.com/MostlyAdequate/mostly-adequate-guide-ru>

The book introduces the reader to the functional programming paradigm and describes a functional approach to developing JavaScript applications. I joined the translation when it got stuck at 60%. Me and my friend refactored every chapter translated before us, and then finished the translation.

JavaScript, Ramda

FP Specialty

From 2019 to 2021 I have maintained a functional programming reading group for people of all functional programming skills.

Education

Mastering Haskell Programming

<https://www.udemy.com/certificate/UC-DRMAMOQ5>

Packt, 2019

Functional Programming in Haskell, part 2 (certificate with honors)

<https://stepik.org/cert/207739>

Stepik, Computer Science Center, 2019

Functional Programming in Haskell, part 1 (certificate with honors)

<https://stepik.org/cert/196007>

Stepik, Computer Science Center, 2019

Computer Science Summer School, Theory of Programming Languages

Novosibirsk State University, 2019

Maintenance of computer equipment and computer networks

Novosibirsk Aviation Technical College, 2004 — 2008

Skills

Haskell

Effects and Streaming: async, conduit, mtl, parallel, resource-pool, retry

HTTP and Networking: http-conduit, servant-server, ua-parser, wai, wai-extra, warp

Testing: generic-random, hspectest, hspectest-golden, QuickCheck, testcontainers

Other: aeson, fmt, haskell-to-elm, lens, nonempty-containers, optparse-applicative, password, pcg-random, postgresql-typed, pretty-simple, relude, safe-decimal, text, uuid, zlib

Scala

Effects and Streaming: akka, akka-stream, cats, cats-effect, fs2, zio, zio-streams

HTTP: akka-http, caliban, finagle, tapir, zio-http

Testing: scalacheck, scalatest, specs2, testcontainers, zio-test

Other: chimney, circe, enumeratum, flyway, quill, scala-parser-combinators, scodec

Elm

UI: elm-charts, elm-css, elm-datetime-picker, elm-style-animation, elm-ui, elm-ui-dialog, svg

Testing: elm-test, elm-review

Other: remotedata, elm-all-set, elm-round, elm-stat

TypeScript

Functional programming: fp-ts, io-ts, rxjs, sanctuary-js, ramda

Network: Socket.io, Apollo

Testing: Jest, Mocha, Jasmine

Frontend: React, Angular, Next.js, Redux, Redux-saga, Vite, Webpack, ESLint, SCSS, Emotion

Other

Databases: PostgreSQL, MySQL, MS SQL, Greenplum, Redis, MinIO, MongoDB

Infrastructure: Kubernetes, Docker, Zitadel, Keycloak, Grafana, Loki, NGINX, Traefik

Testing: Cypress, K6

CI and other tools: GitHub Actions, Just

APIs: GraphQL, OpenAPI

Project Management and Documentation: GitHub Projects, Figma, Draw.io

Soft Skills: research and analysis, project management, code review, leading, mentoring, conducting technical interviews, preparing and giving presentations, running meetup groups and reading groups, initiative, transparency, conflict resolution, providing feedback, active listening