



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 real-time retail solutions, including the Samokat food delivery app and other marketplaces.

I develop data pipelines using platform tools.

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 designed for trading on the Binance crypto exchange. It incorporates a trend-following trading algorithm, an exchange simulation engine for backtesting, an exchange API client for live trading, and a web application for inspecting, debugging the trading algorithm, and displaying statistics.

We employ various testing techniques, including snapshot testing, property-based testing, and case-based testing. Our approach includes schema-first typesafe SQL queries to PostgreSQL. Additionally, we utilize the Haskell-to-Elm library to generate Elm code from Haskell. Also, we have a strong focus on the absence of partial functions. This technological combination enables us to catch most regressions at compile-time.

The project consists of approximately 17,000 lines of Haskell code and 10,000 lines of Elm code. During a technical interview, I can provide a demonstration of the project running in a production environment and highlight relevant sections of 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

As a team lead, I managed two projects, overseeing the implementation of new features, writing tests, and addressing bugs.

Additionally, I mentored new developers, predominantly with a Java background, to familiarize them with Scala. I also organized and led 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

Frontend Engineer

Pamir

05/2020 — 12/2020

I developed a web application that employs server-side rendering and implemented comprehensive unit test coverage. I containerized the application using Docker and established continuous integration (CI).

Additionally, I provided mentorship to 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

Backend Engineer

Eldis

<https://eldis.ru>

10/2019 — 12/2019

I developed a declarative decoder for the internal binary document format, ensuring comprehensive test coverage, 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, addressed defects, and developed new applications and microservices, ensuring coverage with both 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 combination of a loyalty platform, social network, and online store.

In my role, I focused on tasks related to front-end business logic. I actively contributed to enhancing code quality by covering existing code with unit tests and fine-tuning them. This effort resulted in the successful integration of tests into the CI pipeline, effectively preventing defects associated with unsuccessful Git branch mergers in our extensive codebase.

Responsibilities: engineering, testing, code review

Frontend: TypeScript, Angular, RxJS

Frontend Engineer

Allmax

<https://savl.com>

11/2017 — 08/2018

I contributed to the Savl project, a mobile application functioning as a wallet with support for several cryptocurrencies. My primary focus was on the data layer within the mobile application. Applying principles from functional programming and software design, I implemented changes that enhanced the application's modularity and fault tolerance, effectively preventing crashes due to exceptions or unexpected behavior from external services. Furthermore, these modifications facilitated the seamless enabling and disabling of support for individual cryptocurrencies, coupled with the implementation of comprehensive unit testing.

Within the company, I conducted 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, constructed in a code-first, typesafe, and functional way. The servant-to-elm tool generates Elm types, decoders/encoders from Haskell types, and Servant definitions. This not only detects regressions during compile-time but also provides customizable Elm functions for fetching 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, developed with a schema-first, typesafe, and functional approach. Leveraging the mu-haskell library, it performs type checking of Haskell code against GraphQL schema during compile-time. Additionally, the elm-graphql library generates Elm types, decoders/encoders from GraphQL schema. I revamped the Elm frontend and made minor adjustments to the Haskell backend, and uncovered some bugs in the mu-haskell library itself in the process.

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>

Translated the Mostly Adequate Guide to Functional Programming in JavaScript into Russian. The book introduces the functional programming paradigm and outlines a functional approach to JavaScript application development. I joined the translation effort when progress was at 60%. Collaborating with a friend, we refactored each previously translated chapter before completing the translation.

JavaScript, Ramda

FP Specialty

Maintained a functional programming reading group from 2019 to 2021, welcoming individuals of all functional programming skill levels.

Education

Mastering Haskell Programming

<https://www.udemy.com/certificate/UC-DRMAMQ5>

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, hspect, hspect-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