



# Alexander Ryzhikov

🌐 cooba.me 🔄 coobaha 🌐 ryzhikov ✉ coobaha@gmail.com 📍 Finland, Helsinki

## Technical Skills

Languages	JavaScript TypeScript Elixir Go Rescript ReasonML Flowtype OCaml Python Ruby
Backend	Express Fastify Phoenix Ash GraphQL Hapi Meteor
UI Libraries	ReactJS Svelte SolidJS VueJS AngularJS CycleJS EmberJS
Web Frameworks	Remix Astro LiveView Electron NextJS SvelteKit NuxtJS
Mobile	React Native Flutter Cordova Ionic
Databases	Postgres Redis MongoDB CouchDB PouchDB
Cloud	Cloudflare AWS Hetzner Heroku DigitalOcean
CI/CD	Github Actions KamaL OpenShift Jenkins TravisCI CircleCi Gitlab
Observability	Sentry New Relic Datadog Prometheus Grafana
Containers	Docker Kubernetes OrbStack Colima
Testing	Playwright Jest Vitest Cypress Testim.io React Testing Library
Real-time	CRDT Collaborative Editing WebSockets
Job Processing	Oban Kafka
VCS	Jujutsu Git
Developer Tools	VS Code Extensions IntelliJ Plugins
Web Automation	Playwright CDP Browser Extensions Selenium
AI	Claude Code MCP AI Agents Structured Output Prompt Engineering

## Professional Experience

### Principal Product Engineer at Framer [↗](#)

November 2023 — Present

Built a tool for static analysis of the codebase to generate optimal code and libraries based on type information and data structures, resulting in a 30% faster startup time and 20% faster rendering time for end users. [AST](#) [TypeScript](#) [NodeJS](#)

Contributed to CRDT-based collaborative editing system for Framer Studio — binary protocol, columnar storage, and conflict resolution across frontend, NodeJS tooling, and Go backend. [TypeScript](#) [Go](#) [CRDT](#)

Made massive projects (2+GB) work smoothly in Framer Studio [NodeJS](#) [TypeScript](#) [Go](#) [ReactJS](#)

Speeded up loading times of projects for all users by 3–5x (from 20s to 5s) on average by converting Framer Studio to async editing and shallow parsing (lazy pages)

[framer-api](#) — API to be able to access Framer projects programmatically. [NodeJS](#) [TypeScript](#) [Fastify](#) [WebSockets](#) [Browser Automation](#) [Redis](#)

[framer-dalton](#) — CLI tool and AI agent skills for programmatic Framer project interaction via Server API. [TypeScript](#) [tRPC](#) [AI Agents](#)

Product work with Team & Community, helping with bugs and doing on-call, hiring

### Senior Staff Engineer at RapidAPI [↗](#)

December 2022 — August 2023

### Staff Engineer at RapidAPI [↗](#)

December 2021 — December 2022

Collaborated on coding and increased team productivity.

Led large multi-team technical initiatives from ideas to delivery.

Enhanced observability across services.

Led micro-frontend architecture and development.

Promoted modern development practices and team autonomy.

Delivered high-leverage services and features like RapidAPI Studio, HTTP Client (Web + VS Code extension), Hub, Design System, API Linting VS Code extension, and more.

Explored and integrated new technologies, tools, and third-party services like Builder.io into our Marketplace and Admin.

### Staff Engineer at Wolt [↗](#)

June 2021 — December 2021

Collaborated on coding and assisted teams with front-end development needs.

Fostered shared code and conventions while aiding the engineering team's growth.

Advocated for modern development practices and team autonomy.

Promoted common components and ensured the delivery of deliverables.

## Senior Software Developer and Team Lead at Wolt [↗](#)

July 2019 — June 2021

Developed full-stack solutions using [TypeScript](#), [JavaScript](#), [NodeJS](#), [Scala](#), [ReasonML](#), [MongoDB](#), [Postgres](#), [Kubernetes](#), [Docker](#), [Kibana](#), [Graphana](#), [Python](#).

Led front-end development and architecture design.

Implemented a UI toolkit used company-wide.

Led a team of 8 developers (5 front-end devs, 2-3 backend devs) + 2 designers + Product Lead.

Hiring and mentoring.

Facilitated team meetings and tasks.

## Senior Software Developer at Zalando [↗](#)

November 2018 — July 2019

Crafted a performance review tool used internally by the entire company.

Led the development and architecture of user-facing parts & services.

Optimized performance across various platforms.

Designed and implemented new architecture for the translation process.

Improved translation coverage and process efficiency significantly.

Assisted in recruitment, onboarding, and knowledge transfer.

[JavaScript](#) [ES2015+](#) [ReactJS](#) [NodeJS](#) [Webpack](#) [Kubernetes](#) [AWS](#) [S3](#) [CloudFormation](#) [Postgres](#) [Ruby on Rails](#) [Python](#) [ReasonML](#)  
[TypeScript](#) [Scala](#) [CSS4](#) [CSSnext](#) [PostCSS](#) [styled-components](#) [GraphQL](#) [zone-js](#) [OpenTracing](#)

## Senior Software Developer at Relex Solutions [↗](#)

June 2016 — November 2018

Developed new features and projects for the supply chain planning system as a full-stack developer [Ruby on Rails](#) [JavaScript](#)

[NodeJS](#)

Architecture and development of interactive dashboards constructor

Transitioned huge legacy [EmberJS](#) codebase to [ReactJS](#) without requiring a complete rewrite.

[ReasonML](#) automation tool for team daily routines

[ESLint](#) plugin to warn about lousy mutation patterns by parsing code AST

Assisted in recruitment, onboarding, and mentoring and acted as a teacher in the developer academy.

[JavaScript](#) [ES2015+](#) [ReactJS](#) [NodeJS](#) [Webpack](#) [Ruby on Rails](#) [Python](#) [React Native](#) [ReasonML](#) [Java 8](#) [CSS4](#) [CSSnext](#) [PostCSS](#)

## Senior Front End Developer at OneTwoTrip [↗](#)

November 2015 — June 2016

Established cross-project frontend architecture.

Developed and launched new applications and projects.

Supported and developed current legacy apps [ReactJS](#), [jQuery](#), [AngularJS](#).

Assisted in recruitment processes.

[JavaScript](#) [ES2015+](#) [ReactJS](#) [NodeJS](#) [Webpack](#) [GruntJS](#) [GulpJS](#)

## Software Developer at Clickavia [↗](#)

February 2015 — November 2015

Developed [EmberJS](#) version of avia application

Engineered [ReactJS](#) isomorphic dynamic packages application

Created [NodeJS](#) services

Implemented dropped basket analysis

Managed DevOps, CI infrastructure with [Docker](#) and [Jenkins](#)

[JavaScript](#) ([ES2015+](#)) [CoffeeScript](#) [AngularJS](#) [ReactJS](#) [EmberJS](#) [NodeJS](#) [Jenkins](#) [Hapi](#) [Webpack](#) [Babel](#)

## Front End Developer at Media5 [↗](#)

March 2013 — February 2015

Developed new projects for the customers

Developed iOS hybrid app, [AngularJS](#) apps

Utilized [NodeJS](#) stack for a front-end workflow

Supported legacy sites and projects of the company

[JavaScript](#) [NodeJS](#) [Cordova](#) [AngularJS](#) [php](#) [git-flow](#) [Twig](#) [ReactJS](#)

## Earlier Roles at Various

November 2009 — March 2013

Freelancing as Front End Developer (January 2012 — March 2013)

Customer Service Officer at Raiffeisen, Russia, Moscow (August 2011 — February 2013)

Sergeant in the Russian military at Air Force radio-technical troops (June 2010 — June 2011)

Sales Manager at Pixel24.ru, Russia, Moscow (November 2009 — June 2010)

## Framer Studio [↗](#)

Collaborative design tool with real-time editing, CRDT-based document sync, and massive project support

Static analysis tooling for optimal code generation, 30% faster startup and 20% faster rendering

Async editing and shallow parsing for 3-5x faster project loading

TypeScript React Go CRDT Binary Protocols WebSockets

## framer-api [npm](#)

API to access Framer projects programmatically

WebSocket server managing headless browser instances

TypeScript Fastify WebSockets Playwright Redis

## framer-dalton [npm](#)

CLI tool and AI agent skills for programmatic Framer project interaction

TypeScript tRPC AI Agents

## Automated Trading Platform

Automated analysis and execution platform with real-time market data processing

Browser extension integration with remote Playwright script execution for data capture and automated actions

Distributed Elixir cluster with dynamic per-user job queues, circuit breakers, and LLM-powered decision making

Elixir Phoenix Ash Oban PostgreSQL Playwright Docker Prometheus Grafana

## Rapid HTTP Client [↗](#)

A full-featured HTTP client that lets you test and describe the APIs you build or consume. Rapid Client has a beautiful native macOS interface to compose requests, inspect server responses, generate client code, and export API data.

Web version of native macOS app

Every project is a git repo with branches and commits

Offline first (IndexedDB)

Realtime low-latency sync (Firebase)

Extendable with extensions (like code generation or new dynamic values)

Electron and VS Code extension

Concept of DynamicString and DynamicValue with async recursive evaluation

Most components are deeply editable in place (headers, query params, body, etc.)

Has a standalone Client component used in other products (like RapidAPI Testing)

## RapidAPI Studio [↗](#)

A tightly-integrated set of tools an API provider uses to build, test, publish, and manage high-quality APIs.

Micro-frontends architecture

Squads distributed around the globe with frontend apps for different parts of the product

Simple and complex apps with their monorepo codebases and different kinds of publishing

Vue React VanillaJS VS Code extensions Electron

GitHub Private registry Azure for Windows builds

## Internal tools for Wolt [↗](#)

Internal tools for Wolt employees

All kinds of dashboards and tools for different teams

20+ teams contributing to the codebase

Tables, scheduling, maps, etc.

## RELEX Solutions [↗](#)

One platform to unify supply chain & retail planning.

10+ years project, led for three years, 5-10 core frontend devs and 50 active contributors

Migration to React from Ember

Own query language with different editors and views, big data

Tableau-like highly customizable user dashboards with drag-and-drop panels

Panels are interactive inputs for other panels and produce outputs

Recursive drill-down into other dashboards

## Toy-clone of LogRocket

Clone of LogRocket, slightly similar to replay.io

Full stack ReasonML (now Rescript)

Node.js Kafka PostgreSQL React ReasonML Websockets Docker

Session player with seeking drag-and-drop

---

### *Education*

Specialist degree in Economics, World Economy — Moscow State University of Economics, Statistics, and Informatics (2010)

---

### *Other*


Developed Electron application to automate accounting routines


Built Telegram bot for team daily operations (e.g., assign code reviewers)

Browser Session replay (LogRocket/Sentry clones) — ReasonML, Postgres, Kafka, Protobuf, Redis

 typed-fastify

 jjtask

 suomea

 graphql-to-reason

 postcss-variables-loader

---

### *Languages*

**English** — Full professional proficiency

**Russian** — Native

**Swedish** — Limited working proficiency

---

### *Certificates*

**Algorithmic Toolbox** — University of California, San Diego & Higher School of Economics on Coursera. April 10, 2016

**Data Structures** — University of California, San Diego & Higher School of Economics on Coursera. May 9, 2016