Artyom Suchkov

Senior backend developer

Prague Node.js Developer with 7+ years experience on different projects like FlowReader, WebExpo, OddsPortal, Livesport API, Semrush (My Reports, Client Portal). Having participated in developing more than 35 applications, I'm constantly learning new things and following new trends & technologies. I'm quick learning and I have an ability to adapt. I have experience of Junior & Middle Developers tutoring and Senior Developers on-boarding processes.

LinkedIn: linkedin.com/in/artyom-suchkov

Legal status

I have Belarus Republic citizenship.
I have a permanent residence in the Czech Republic and now I'm learning distance as a software developer.
I'm a tax resident of the Czech Republic.

Work experience

1) **Independent Java Developer** (from 1.2012 to 6.2014)
   Minecraft.su
   Minsk
2) **Middle Software Developer** (from 6.2014 to 5.2016)
   Lead Software Developer (from 5.2016 to 11.2017)
   Wikidi a.s.
   Prague
3) **Senior Node.js Developer** (from 12.2017 till 07.2022)
   Livesport s.r.o.
   Prague
4) **Senior Node.js Developer** (from 08.2022 till today)
   Semrush CZ s.r.o.
   Prague

Publications

- How to trace the history of MySQL data (WebExpo, Prague, *4.5 stars*)
- Reverse engineering of Windows Phone diagnosis application (Habrahabr, *65 upvotes, 5.9K views*)

Objectives accomplished

Wikidi:

1) **Switching from vanilla JavaScript to React as a junior full stack developer:**
    - Writing React + Flux + Immutable.js code in ES6
    - Writing PHP code for REST API (with Nette)
    - MongoDB analytics and optimization
    - Adding rate-limiting functionality with use of Redis
2) **Adding new social network connections (API and OAuth):**
    - Writing React + Flux + Immutable.js code in ES6
    - Writing PHP code for REST (with Nette framework)
    - Work with external API and OAuth (Twitter, YouTube, Google, Pinterest, Reddit, Pocket, Facebook)
3) **Refactoring of PHP queues to Node.js microservices:**
    - Writing Node.js code
    - Work with MongoDB (aggregation framework, map-reduce)
    - Work with Redis
    - Performance boost of news distribution up to 115%

Livesport:

1) **Analysis and fix of MongoDB solution, data migration without data loss on production:**
    - Analysis of existing solution on the developer side (used libraries, type of requests, the data structure)
    - Analysis of an existing solution from DevOps side (replication, the cardinality of keys etc)
    - Creating a team and leading two small projects that are required to migrate data
    - Creating a microservice for data migration without data loss
2) **Creating and supporting API in Node.js**
    - Optimizing requests to MongoDB and MySQL
    - Creating GraphQL schema
    - Writing client-side code for GraphQL
    - Writing beautiful documentation in English (even with pictures)
3) **Migration and modernization of legacy API from CoffeeScript to TypeScript**
    - Switching from CoffeeScript to TypeScript (50 thousands lines of code)
    - Switching from DI containers to ES6 modules
    - Switching from callbacks to async/await
    - Simplification of API architecture from 6 dependencies to 0
    - Adding GraphQL layer without deleting the original REST one
    - Performance boost for JSON.stringify in REST API using code generation
    - Switching from own microservices for cache to Redis and ProtoBuf (without split-brain syndrome) - 70 thousands requests per second with medial 5ms (0.95 quantile) vs 25ms (0.95 quantile)
    - ProtoBuf optimization, creation of a library for type generation, migrating cache to ProtoBuf
4) **Streaming of relation database updates**
    - Without data loss (reliable queue model in Redis) - with At Least Once guarantee
    - >99.99% uptime
    - High Availability, Kubernetes, Gitlab CI
    - Made a presentation about it on WebExpo with an average 4.36 stars.
5) **Creating a write API with persistent queue and transactions for multi-database environment**
    - Worked as an project architect, designed and implemented all of the core application logic

- Abstract to the level where database layers can be added separately without any refactoring
- 100% of data that were added to the queue were processed (transaction & rollback garance)
- Based on many microservices that are communicating using persistent queues & message queues
- Technologies that were used: GraphQL, ProtoBuf, ActiveMQ, Redis (and LUA scripts), MongoDB, MySQL

Semrush:

1) **Refactoring Legacy Applications to TypeScript:**
    - Took a role in upgrading our legacy JavaScript applications to TypeScript, making the code more robust and maintainable.
    - Focused on improving code readability and reducing bugs.
2) **Enhancing Application Security:**
    - Focused on identifying and mitigating security vulnerabilities in complex applications, specifically targeting SSRF, XSS, and RCE threats.
    - Implemented best security practices, significantly reducing potential risks and safeguarding user data.
3) **Achievement in Security CTF:**
    - Competed in a challenging security Capture The Flag (CTF) competition and secured 3rd place out of 20 participants.
4) **Full Lifecycle Development with Nest.js, Prisma, TypeScript:**
    - Had a key role in the project from conception to deployment, using a modern stack containing Nest.js, Prisma, and TypeScript, while ensuring high-quality, scalable, and maintainable code.

Pet projects:

1) **USB Report Collector for Hutermann CO2 Meter**:
    - Developed a custom USB report collector for the Hutermann CO2 meter, integrating low-level programming.
    - Achieved real-time monitoring and analysis of environmental data, contributing to efficient workspace management and air quality control.
2) **HSM Implementation for Flipper Zero:**
    - Developed a Hardware Security Module (HSM) and Password Manager application for the Flipper Zero *(a multi-tool device for penetration testers)*.
    - Implemented secure communication using the I2C protocol and an ESP8266 microcontroller, ensuring data integrity and security.
    - Designed both hardware and software components.
3) **[WIP] Open-Source Rotary Evaporator:**
    - Currently developing an open-source Rotary Evaporator, blending hardware engineering with software development.
    - Using C++ and ESP8266, targeting efficiency and user-friendliness in laboratory environments.
    - Utilizing SolidWorks for mechanical design, focusing on reliability and ease of assembly for DIY enthusiasts.
4) **Various Web Scrapers and Binary Protocol Decoders:**
    - Created a suite of web scrapers and binary protocol decoders.

Technology stack

Programming languages:

- **JavaScript/Node.js (proficient knowledge)** - my main programming language. I have a big experience (more than 7 years), mostly with TypeScript. I know everything that is necessary, including the latest EcmaScript features. I can optimize well, I know how prototypes work in V8, what functions are slow and how to make the faster. Can write an effective but beautiful code. I have experience in developing applications of any kind, both backend and frontend. I can understand any code and project. I know many libraries, and I'm able to write production-ready highly loaded microservices with HA, monitoring, logging and tracing. I have a deep knowledge of all the inner parts of the language.
    - **ESLint** - I'm using an Airbnb configuration, used both with TypeScript and JavaScript configuration, and migration from TSLint to ESLint.
    - **Express.js** - experience in developing high-load applications with Express.js, integration with Apollo GraphQL.
    - **Jest/Mocha** - experience with mocks, snapshots, jest.spyOn, as well as with the test optimization. I know how Jest work from the developer side.
    - **React** - experience in creating websites from scratch, experience with Flux and Redux, as well as with Immutable.js.
    - **Redux** - experience with creating applications from scratch and supporting existing ones. Bigger experience probably with Flux (predecessor of Redux).
    - **Apollo GraphQL** - experience Apollo Server and Apollo Client, experience with code generation of GraphQL schema from JSONSchema and other custom formats, experience with both GraphQL object schema and GraphQL string schema, DataLoader and N+1 problem solution, experience with full migration from REST API to GraphQL, dynamic binding of objects and entity relation.
    - **grunt/gulp/webpack** - small experience in different projects.
- **TypeScript (proficient knowledge)** - big experience of creating backend and frontend applications, migration from JavaScript to CoffeeScript, experience with generics, type definition and writing clean code. I carefully follow updates of each version.
- **GraphQL (proficient knowledge)** - experience Apollo Server and Apollo Client, experience with code generation of GraphQL schema from JSONSchema and other custom formats, experience with both GraphQL object schema and GraphQL string schema, DataLoader and N+1 problem solution, experience with full migration from REST API to GraphQL, dynamic binding of objects and entity relation.
- **PHP (advanced knowledge)** - big experience in creating monolite applications, web sites and switching from PHP to JavaScript.
- **C# (intermediate knowledge)** - medium experience of developing different .NET tools on Windows, source code analysis and deobfuscation, I have a small article about that on Habrahabr. I had experience developing an application for Windows Phone (Meridian, Pikabu client).
- **Java (intermediate knowledge)** - pretty big experience of creating applications in the past, mostly for Minecraft game. Can definitely understand code and fix errors. I created a few small plugins for the message queue I use, and made some pull requests to a big repository with error fixing. Experience in creating a small application from scratch using Maven and JUnit.
- **C (basic knowledge)** - experience in the university, source code analysis and writing plugins for Node.js, Redis and microcontrollers.
- **C++ (basic knowledge)** - experience from the university, source code analysis and writing plugins for Node.js.

- **Go (basic knowledge)** - small experience while creating a small application, I understand the syntax and can learn quickly.
- **Python (basic knowledge)** - small experience and good understanding of syntax, can learn quick if needed

Source version control

- **Git (proficient knowledge)** - experience with both Git Flow, and One Flow (I prefer One Flow). I understand git well enough to work with it from the command line and not be afraid of it.

Instruments and services

- **Grafana (proficient knowledge)** - experience with creating a big amount of panels with data from Prometheus, ElasticSearch and InfluxDB. Also with alerts, annotations and variables.
- **Chrome DevTools (proficient knowledge)** - can read waterfall, cpu and memory reports, analyze requests and make scrapers based on data from DevTools.
- **npm/yarn (proficient knowledge)** - creation and publication of libraries, creation of library template, source code analysis (security), error fixing, migration from npm to yarn back and forward. Understanding of semver and the structure of dependencies. Experience with node_modules optimization.
- **PHPStorm/Webstorm/CLion** - using it every day, I'm running TypeScript with debugging, using File watcher for automatization and refactoring is a pleasure for me with Webstorm.
- **GitLab (advanced knowledge)** - using it every day, I have experience with creating pipelines for app/image publishing, deploying with SSH using GitLab, scheduled pipelines and environment variables.
- **Kibana (intermediate knowledge)** - experience in log analysis, creating templates and indexes, and using Lucene language.
- **Visual Studio Code (intermediate knowledge)** - using it sometimes, I can use it if needed.
- **YouTrack (intermediate knowledge)** - using it every day
- **Keycloak (intermediate knowledge)** - small experience in using and configuration, I was doing a research of possibility of using this tool in our projects
- **Icinga (basic knowledge)** - small experience in configuration and usage

Message queues

- **ZeroMQ (advanced knowledge)** - big experience in using this (obsolete) technology in production
- **NATS (advanced knowledge)** - experience in using this technology in production
- **MigratoryData (advanced knowledge)** - experience in using this technology in production and writing plugins
- **ActiveMQ (basic knowledge)** - experience in using this technology in production
- **Apache Kafka (basic knowledge)** - experience in using this technology in production
- **RabbitMQ (fundamental knowledge)** - small experience

Databases

- **MongoDB (proficient knowledge)** - big experience with analysis of performance and creation of report of required actions. Experience with shards, indexes, HA, data loss, replication performance, migration of BigData without losses and stops in production. Experience with requests optimization (write and read), aggregation framework and map-reduce. Experience with OpLog and Change Stream.
- **Redis (proficient knowledge)** - big experience with caching and invalidation, Lua scripts, performance and reliable queue creation. Worked with sentinel and I know how to make cache quick, I know how Redis works. Wrote some plugins for Redis in C.

- **MySQL (advanced knowledge)** - experience in writing queries, indexes. Experience with MySQL 8 problems. Experience in streaming changes from MySQL. Experience with replications and replication lags.
- **ElasticSearch (intermediate knowledge)** - experience of using this technology in production and writing lucene queries, got a course about indexes and performance.
- **Neo4J (intermediate knowledge)** - experience in one-time migration of data from MySQL to Neo4j, and writing different queries.
- **InfluxDB (basic knowledge)** - experience of using this technology in production and writing requests.
- **Sphinx (basic knowledge)** - experience of using this technology in production and writing requests.
- **PostgreSQL (fundamental knowledge)** - small experience in the past.

DevOps

- **Prometheus (proficient knowledge)** - big experience in application monitoring, auto-discovery using Consul.io, wrote a Node.js library for Prometheus that works 55% faster than prom-client.
- **Consul.io (advanced knowledge)** - experience of using this technology in production, service discovery for Prometheus and also for load balancing in pair with Nginx.
- **Docker (intermediate knowledge)** - experience in application containerization, docker-compose, security of containers and publicating containers to quay.io.
- **Kubernetes (intermediate knowledge)** - experience with microservices architecture orchestration, including HA.
- **nginx (intermediate knowledge)** - experience in using this application in production. Experience in using Nginx as a reverse proxy for microservices and also connecting to OAuth service. Experience with load balancing using Consul.io.
- **Apache HTTP (intermediate knowledge)** - experience in using this technology in production.
- **Google Cloud (intermediate knowledge)** - creating infrastructure & deploying applications
- **HELM (intermediate knowledge)** - creating infrastructure & deploying applications
- **Filebeat (basic knowledge)** - experience in writing a high-load logger working on UDP.