Anton Korniychuk

Web-Development expert

Specialization: Angular, NestJS, RxJS

+380 (96) 887 56 17 anton@korniychuk.pro
t.me/korniychuk_pro www.korniychuk.pro

Knowledge of technologies and software:

Updated April 9, 2022
the latest revision at www.korniychuk.pro/cv

js	Angular 2-13, RxJS 4-7, NgRx, Material UI, PrimeNG, NgBootstrap, AngularJS 1.x, ES5 , ES6/7...12, TypeScript 1/2/3/4 WebSocket, XPath, Writing Chrome Plugins,
js testing	Jest + writing custom rules for Jest, Wallaby.js/Quokka.js, Codelyzer/AngularESLint, Storybook, Spectator, RxJS marbles, Karma, Chai, Jasmine, Supertest, Protractor, faker.js, TDD, Gherkin Notation
js libraries	DayJS, MomentJS, LoDash, LoDash FP, Ramda, class-validator, class-transformer, ts-toolbelt, utility-types, Ag-Grid, jQuery, D3, ccxt, Neon (Rust NodeJS extensions)
js backend	NestJS 8, ExpressJS, Fastify, Koa2, SocketIO, pm2, caporal, TypeORM, Prisma, Worker Threads/Child Process
layout	html 5, css 3+, flex-box, grid layout, sass/scss, less Bootstrap 3/4, Animate.css, Font Awesome, Admin LTE, Autoprefixer, BEM, SMACSS, Atomic Design
bundling	Webpack 1/2/3/4, Babel, Grunt, Gulp
code-quality	ESLint, TSLint, CSSLint, StyleLint, Prettier JSDoc, TSDoc, CompDoc, writing custom rules for TSLint & ESLint for TS
databases	PostgreSQL, TimeScaleDB, MySQL, MongoDB, SQLite, Redis, Firebase
queues	Rabbit MQ, Amazon SQS, Redis
design	DRY, KISS, YAGNI, GoF, SOLID, GRASP, TDD 12 factor apps Now learning: DDD, Clear Architecture, Rich Models
php	PHP7/5.6, PHPDocs, PHPUnit, Composer, Doctrine, Codeception, Symfony 3, Laravel 5, Kohana, Yii2, Phalcon, Smarty

vcs	Git (github, bitbucket, gitlab, gitea), git-flow, SVN
monorepos	Npm, Yarn 1/3, Lerna, NX
ci/cd	GitHub Actions, BitBucket Pipelines, GitLab, TeamCity, Jankins, Travis CI, Circle CI
api docs	Swagger, Postman, raml, json-schema
editors	NeoVim, Vim, WebStorm, DataGrip, PhpStorm, PyCharm, Sublime Text 3, Atom, MySQL WorkBanch, diagrams.net
tasks	Trello, Jira, Redmine, ClickUp, Asana, Confluence, Slack, Zulip
os	AlpineLinux, CentOS, OpenSUSE, Debian. RouterOS/Mikrotik (basic knowledge)
servers	Nginx, Apache 2, OpenSSH, Bind(dns server), OpenVPN, ProFTPD
shell	bash, zsh, sh, grep, find, awk, cron, vim
virtualization	Docker/Docker Compose, VBox, Vagrant, Parallels
graphics	PhotoShop, Illustrator, SketchUp Pro, GIMP, FFmpeg/libav
other	I'm a RexExp expert. SSL, Let's Encrypt, Verdaccio (private npm), Graylog basic krowledge of Rust & writing NodeJS plugins, GCM, APNS, SSH, FTP/SFTP, GCC/Make,
notes	Obsidian.md, Boostnote, Notion

My benefits:

- I work with Angular 2+ since the first beta version (January 2016). Already 6 years.
- Over 10 years of web development experience.
- I like things that most people hate!
  - Strict 100% TypeScript typing
  - RxJS Marbles
  - Using Vim & Vim-mode in IDE
  - RegEx, grep, parsing, etc
  - using shell & Bash scripting
  - Configuring WebPack & Dockerfile
- I'm a Regular Expressions expert.
- Experience in teaching and training developers.
  - Speaking at conferences.
  - lots of internal tech-talks in companies about Angular, RxJS, RegExp, Bash.
- Confident blind typing in Russian and English (~350 chars/min).
- Over 7 years of experience with OS Linux.

Professional and personal qualities:

- I note everything, as the result forget nothing.
- I love order. I never have 100 open tabs in chrome.
- I consider myself a not conflict person.
- I know how to present my views and sell my ideas.
- I'm not afraid of public speaking.
- I'm a team worker, and relationships are important.
- I'm result-oriented, disciplined, persistent, and purposeful.
- My English is B1.

Work experience:

Fullstack Developer
Company name is under NDA

July 2021 — Present

Cryptocurrency algo-trading company. Most of the time I worked on the backend side with NestJS and TimeScaleDB. Lots of domain logic was written in pure TypeScript. Code was organized as a big monorepo managed by NX.

Backend:

- NestJS, TypeORM, and Prisma in some microservices, class-validators, and class-transformers. Postgres and TimeScaleDB as the main databases. Redis as cache.
- Backend was implemented as a bunch of microservices. Some of them were located on AWS & Hetzner clouds, some on simple servers with 16 volumed HDDs in RAID.
- Raw TCP and Amazon SQS for microservices communication.
- OrderBooks real-time analysis was implemented as Rust extensions for NodeJS. They were connected to crypto exchanges directly, via WebSockets, listening data, analyzed it, and on so-called signal is detected - some information was sent to NodeJS via Neon.
- Rust extensions were used for better performance, economy RAM use, and multithreading.
- This system saves an extremely large amount of markets data every day. ~600GB of uncompressed database records

being stored to distributed tables in TimeScaleDB every day.
- Graylog as the logs management system.
- In this project, I tried to write business logic as technologies-agnostic as it was possible. It wasn't full-featured DDD, but we tried to have Ubiquitous Language, split business domains by contexts, and implemented Rich Models in some microservices. We tried to make domain logic a core that is independent of anything.

UI on this project was not so big. It consisted of 3 parts.
1. Reports were implemented via AgGrid + Material UI.
2. Admin panel to manage the trading algorithms: Material UI.
3. Visualizing market data, opened and closed positions, orders etc.
In this UI part performance was so critical. All data was transferred via WebSockets. We followed every change detector call very strictly. Most of the components were dumb, in lots of components automatically change detection was disabled. All calculations and data operations were computed by web workers. The main thread was only for drawing. In this part of UI I've implemented an interesting feature name order book heating map. It was a custom full-screen chart implemented using D3 that shows everything in the order book of some trading pairs under a microscope.

Angular Developer
Dev.Pro company

July 2018 — June 2021

I worked on a big enterprise system in the Point of Sale sphere. The system includes lots of UI projects, integrated with each other, micro-services, and custom UI/Backend libraries. More than 150 git repositories. Most time I worked on 2 projects.

Project 1: Food ordering browser app.
- ~9 people team. I was leading the UI side.
- Used libs: NgRx, Angular Material & SDK.
- Lots of Integrations. With other sub-project, with payment systems (GPay, ApplePay, Heartland), many food delivery services, google maps.
- Complex business logic on the UI side. The backend aggregated and proxied data, and provided some payment features.

Project 2: Hug admin panel for restaurants.
- ~25 people team. Libs: Angular, angular-redux, AgGrid, LoDash FP.
- ~200 separated business entities (domain models).
- If you can imagine how many pages an admin panel should have to configure ~200 entities, imagine that half of them contain more than one complex grid (ag-grid) on the page.
- There were ~7000 unit tests on the UI side.
- At least 3 entities had so-called 'formula' field. Those were fields that container expressions in a simple meta-programming language that we developed. Then these expressions were consumed and executed in different

environments (python, sql, js, etc...). In our UI there was implemented a convenient formula editor for administrators that didn't have coding experience.
- There was a period with big epics (root level tasks). For example, there was an epic that was estimated in ~700 man-days.

Both projects:
- Had very complicated business domain/rules.
- I had an experience with upgrading Angular from v2 to v11 in the first project and from v4 to v9 on the second one. It was difficult but interesting. I faced lots of tricky, and not documented breaking changes inside Angular CLI (Webpack), Ag-Grid, RxJS, NgRx, Angular itself, and typing of libraries like LoDash.

Team Lead
Unicsoft company

February 2017 — July 2018

The project is an information system, in which there are a lot of statistical information, tables and graphs. Little input and much output. The challenge before us was to recreate a fairly old project on Angular 2+, which was written in ActionScript 3 (Flash).
When I came to the project, it was already in development for almost a year. The project was in a difficult state due to the chaos in the code and a very large technical debt. The architecture was completely unintelligible. One of my tasks was to gradually refactor the entire application. Gradually, the code managed to lead to a structure similar to the Angular application.
The project contains a lot of data and the tables contain 10k lines here in the order of things. For working with tables, I suggested using a commercial version of ag-grid and it showed itself perfectly. The selection criteria were: configurable, productivity and the ability to export to excel and pdf. Out of the box was the export only in csv and excel. Export to pdf we implemented with the help of a third-party library.
One of the problems on the project is boot time and performance. This is something we are working on right now. The whole project was developed as one single module, which is loaded immediately on page loading and it takes very long time. The connectivity of different parts of the system is very high.
Now we are engaged in sawing the system into modules. We are moving towards replacing gulp build system to webpack. Gradually, in small steps, we cover the system with unit tests.

Web-developer
Adoriasoft company

March 2016 – February 2017

There was a large and very old project - cloud storage with more than 10,000,000 users. But it was not a high load. We did it support and the addition of new features. This project originally was written on php 4. Then php has been upgraded to version 5.3. This project does not use any frameworks,

only Smarty Template Engine. Interesting problems:
- When I started work, there was the image of the virtual machine takes more 40GB. We did not like it and the first thing we did is to recreate the entire environment from scratch on Vagrant.
- The project lay on the SVN repository, and has about 30 branches, which had been abandoned and no one knows where it is and what to do with them. We have moved some of the most important branches on bitbucket. All the commit history has been preserved during migration.
- Introduction of OOP, namespaces, unit-testing and continius integration using TeamCity.
- Replacement of the payment system, as well as the establishment of the level of abstraction for the quick-change payment system. The work was carried out with the use of TDD with more than 80% of the test coverage.
- I implemented webpack 2 to build typescript and sass and many other things.

There was also several smaller projects. I managed to work with Laravel, WordPress, Angular 1/2.

Another interesting project was a single-page site. This is business card site that I designed on Angular 2 and NodeJS. An interesting feature of this site has been a huge number of complex and dynamic animations including embedded svg. All transitions have been implemented in the css and Web Animations API. 80% of the work time in this project was spent on animations. The animations was adapted for all screen resolutions.

Web-developer
OnCreate company

August 2014 – present

Admin panel, RESTful api.
We work on our startup project. Project has mobile Android and iOS application. I'm writing a server part. One of the interesting features of the project - extensive statistics, about 100 meters, it operates using redis db.

Lecturer
STEP IT Academy

july 2014 – February 2016

Chastised lectures on backend and frontend development of more than a dozen groups.

Mainly were evening groups. Also, two groups of children were.

Periodically hold it-events the topic of new technologies. At the beginning of 2016 I held two it-event on the theme of «Ecma Script 6» and «Angular 2.x».

Software for Yan Daniloff company

january 2013 — july 2013

Closed software for apparel companies. Order accounting, wages, price list, reports in excel etc.

Technologies: js, jQuery, mysql, php, smarty
The architecture is built so that all requests to the database are performed through stored procedures.In this project, I

have written more than 150 stored procedures for mysql.

The company has outdated computers so we had to do a lot of optimizations and hacks in the JavaScript that have all worked quickly.

Freelancer

september 2012 – march 2016

Admin panels, RESTful api, I started on pure php, then Kohana 3.3, and from mid-2015 Yii2.

Online Stores based on OpenCart 2 and ShopScript5

At the beginning made small sites on wordpress and joomla.