Anton Grishin — Python Developer

Personal Information

Moscow, Russia
○ alchemmist
in anton-grishin
✈ @alchemmist
☎ +7(915)067-2638
✉ anton.ingrish@gmail.com

About Me

I am a student and aspiring developer. I have been programming for 4 years, during this time I have contributed to 44 repositories, made 921 commits, and written 176,729 lines of code. I teach Python, having trained over 30 students. Co-founder and developer at Ballkit. Previously a professional volleyball player.

Education

Central University — Mathematics and Computer Science, 2028 *(1st year)*. Major: "Development". Track: "Computer Science Research".

Skills

Python: `FastAPI` , `pydantic` , `Flask` , `SQLAlchemy` , `faststream` , `alembic` , `pytest` .

Databases: `postgres` , `sqlite` , `redis` , `Yandex Object Storage` .

Message brokers: `RabbitMQ` , `Mosquitto` .

Other technologies: `SQL` , `Java` , `JavaScript` , `bash` .

Dev tools: `Docker` , `Podman` , `Make` , `CI/CD` , `Linux` , `Git` .

Achievements

Reached the **final** (1) of the Russian Championship in sports programming, among 5000 participants. Developed a microservice architecture of 9 services for the web application aggregating sports events across Russia. And developed an algorithm for processing and validating annual government reports on sporting events. Tech stack: `Kafka` , `React` , `RabbitMQ (FastStream)` , `FastAPI` , `OAuth` .

Winner of the "Science for Life" scientific-practical conference among 117 projects with a smart home system project for private and public educational institutions. Tech stack: `Redis` , `Zigbee2MQTT` , `websockets` , `Go` , `Python` , `Flask` , `React` .

Third Prize winner, MPGU Informatics Olympiad: "Applied Informatics"

Certificates

Yandex Lyceum. Entered the 2nd year of the Yandex Academy Lyceum in "Industrial Development" as an external student. Solved 236 problems, implemented and defended 3 projects, final scored 100/100. Graduated with a diploma with honors.

Personal Projects

SmartCab Smart home system for schools   September 2023 — May 2024

Developed design prototypes. Implemented them (1, 2) in `React` . Wrote a bot for system administration. Implemented interaction with 5 smart devices via the Zigbee protocol through Mosquitto broker, and 3 scenarios in the classroom. Tech: `Python` , `Go` , `JavaScript` , `bash` , `Flask` , `sqlite` , `redis` , `postgres` , `JsSip` .

Starship Text-based action game on Telegram   March 2023 — April 2023

Developed mechanics for online duels with real players in a Telegram bot. Also created a flexible message rendering engine using jinja2 templates to handle numerous scenario and environment texts. Tech: `Python` , `Flask` , `alembic` , `python-telegram-bot` , `SQLAlchemy` , `jinja2` .

Corpse Game engine   November 2022 — January 2023

As part of a team, developed a fully functional modular game engine based on the ECS architecture. Also created a survival top-down shooter game set in a zombie apocalypse. Tech: `Python` , `Pygame` , `esper` , `pytmx` , `aseprite` , `Tiled` .

Telegram Bots *All source code publication approved* *Summer 2024*

- Developed a bot for administering and selling WireGuard keys by subscription. Implemented clear analytics for clients directly in the bot. Tech: `Python` , `sqlite` , `python-telegram-bot` .

- Bot for aggregating relevant messages from chats, channels, and groups by keywords. Tech: `Python` , `theleton` , `redis` .

Other Projects *All source code publication approved* *Summer 2024*

- Parser for the Portuguese embassy with authorization (1), captcha solving (2), and branching form traversal (3). Created a Docker image for containerized parsing with `selenium` under Chromium. Configured pre-commit hooks for `ruff` linter and `pyright` static analyzer. Tech: `Python` , `selenium` , `psycopg` , `SQLAlchemy` , `FastAPI` .

- CLI tool for secure operations on Outline servers for thousands of clients. Tech: `Python` .

Formal Verification: I am taking the Software Foundations course. I have read the first volume and proved 250 theorems in `Coq` .

Linux: I use Arch with the Hyprland compositor. I have published my `.dotfiles`. I wrote my Neovim config and a color theme from scratch. I have written over 20 custom scripts.

→ more projects on GitHub