

Чумаков Максим

Python-разработчик (Django / FastAPI)

Москва · Гибрид / офис / удалёнка · chumakov.mr@gmail.com · Telegram @k1y0mi

GitHub: github.com/k1y0miiii · Портфолио: k1y0miiii.github.io · Hugging Face: huggingface.co/k1y0mi

О себе

Backend/full-stack разработчик на Python. За год коммерческой разработки в одиночку спроектировал и довёл до прода четыре мультитенантные платформы — от схемы БД и API до деплоя и нагрузочного тестирования. Работаю с асинхронным стеком (FastAPI, async SQLAlchemy, aiogram 3), интегрирую LLM (Claude API, self-hosted модели), администрирую парк из ~30 Linux-серверов. Привык отвечать за продукт целиком: внедрил Git и нагрузочное тестирование (k6) там, где их не было. Ищу позицию middle backend в команде с живыми процессами.

Ключевые навыки

Языки: Python, SQL, JavaScript, Bash

Backend: Django 5.x, DRF, FastAPI, aiogram 3, SQLAlchemy (async), REST API, вебхуки

Базы данных: PostgreSQL, Redis (кеш, очереди, shared state)

Инфраструктура: Docker, docker-compose, Linux (Ubuntu/Debian), nginx, gunicorn, systemd/supervisor, Git, VPS, DNS/SMTP

Нагрузочное тестирование: k6 (smoke/ramp/stress/soak/spike, xk6-sql), профилирование запросов

AI/LLM: Anthropic Claude API, self-hosted LLM (OpenWebUI/Ollama), fine-tuning (ruBERT), LLM-as-judge, песочница исполнения кода (Judge0)

Frontend: Tailwind CSS, Bootstrap, Alpine.js, vanilla JS/AJAX, Chart.js

Опыт работы — 1 год

Full-Stack разработчик (контракт) — Nik-A Tech (быв. iQnix Tech), удалённо май 2025 — н. в.

- Единственный или ведущий разработчик четырёх продакшн-платформ, а также CMS компании и официального сайта; полностью принял задачи ушедшего разработчика.
- Внедрил Git в кодовую базу, которую до этого правили напрямую на проде; ввёл нагрузочное тестирование (k6) как практику перед релизами.
- Администрирую ~30 Linux-серверов клиентских развёртываний: провижининг, деплой, DNS/SMTP.

СОР-УПК — мультитенантная SaaS приёмной кампании · Django, PostgreSQL, Redis

- Спроектировал и написал в одиночку за ~2,5 месяца к началу приёмной кампании; 200+ роутов. Сейчас используется 4 вузами/колледжами, 20–30 сотрудников, 500+ заявлений за кампанию. Поддерживаю единолично.
- Полный цикл приёмной кампании: RBAC на 5 ролей, многошаговые формы с условной логикой, изолированный деплой под каждую организацию.
- Конструктор генерации документов DOCX: организация загружает свои шаблоны и маппит плейсхолдеры через UI настроек — без участия разработчика.
- Модуль онлайн-экзаменов: песочница исполнения кода Judge0 + LLM-проверка решений на self-hosted моделях (OpenWebUI): подбор моделей, LLM-as-judge.

MyDNS — распределённый сервис проверки DNS-пропагации · Django, PostgreSQL

- Master-master репликация на 4 гео-узлах без брокера: журнал событий на сигналах, идемпотентность по UUID, разрешение конфликтов last-write-wins, курсорная синхронизация, HMAC-аутентификация узлов.
- Параллельный резолвинг через 25 мировых резолверов с картой на d3-geo.

MAX — шлюз интеграции с мессенджером MAX

- Единая точка интеграции с мессенджером MAX для всех проектов компании. Self-service подключение: создание бота, выдача API-эндпоинта с инструкциями; маршрутизация, доставка и централизованный трекинг сообщений.
- Мультитенантность: хеширование ключей клиентов (SHA-256), HMAC-подпись коллбэков, очередь повторных доставок.

СОР-SAO — платформа экзаменов и тестов (спин-офф экза-модуля СОР-УПК)

- Отдельная система, заточенная только под проведение экзаменов и тестов. Собственный пакет профилирования `django_perf`; решал миграционные кейсы (NULL-distinct в PostgreSQL) и рассинхрон состояния под multi-worker gunicorn.

Открытый код (личные проекты)

- [ru-ai-text-detector](#) — детектор AI-текста на русском: fine-tuned ruBERT, калибровка $FPR \leq 3\%$; [live-демо на Hugging Face](#). PyTorch, transformers.
- [token-diet](#) — живой плейграунд: во сколько токенов и денег обходится JSON по разным моделям; на воспроизводимом бенчмарке JTF (~40% без потерь). JavaScript, Python.
- [json-token-format \(JTF\)](#) — компактный lossless-формат JSON для LLM: ~33% меньше токенов, обратимый. Python.
- [glyph-mcp](#) — MCP-сервер: web search и fetch чистым текстом для LLM-агентов, ~8× меньше токенов на запрос. JavaScript.
- [glyph](#) — веб-браузер в терминале: HTTP(S), парсинг HTML (html5ever), подмножество CSS, табы, формы. Rust.
- [ids-nsf-kdd](#) — ML-система обнаружения сетевых вторжений на бенчмарке NSL-KDD: RandomForest / GradientBoosting / LogReg, бинарная и многоклассовая классификация. Python, scikit-learn.

Образование

Институт деловой карьеры, Москва — бакалавриат, «Прикладная информатика в экономике», выпуск — июнь 2026.

Дипломный проект: **SCBP** — веб-система расчёта стоимости проекта (Django): параметрические модели оценки, расчёт по формулам, формирование смет.

Условия

Формат: гибрид (Москва) / офис / удалёнка · Занятость: полная · Оформление: ТК / ГПХ / самозанятость — любой вариант