

РЕЗЮМЕ

Контакты

ФИО	Турбинов Сергей Романович
Город / формат работы	Москва · remote / hybrid
Email	turbinoff@ya.ru
Telegram	https://t.me/freems
GitHub	https://github.com/freemspwnz
Сайт	https://freems.ru
Желаемая позиция	Python Backend / Platform Engineer

Профиль

Python Backend / Platform Engineer: проектирую асинхронные Python-системы и довожу их до production — границы слоёв, устойчивость внешних API, мониторинг и воспроизводимый релиз.

С 2025.12 перешёл в backend разработку. За это время самостоятельно довёл до production полный цикл: backend Telegram-бота, open-source библиотеку circuit breaker на PyPI, клиент encrypted-бэкапов и SSR-сервис с HTTPS. До этого — бизнес-анализ (см. ниже).

Сильные стороны: async Python, retry / circuit breaker / rate limit, метрики и логи, CI/CD и Docker.

Навыки

Языки и runtime: Python 3.10–3.13 · asyncio · Bash 4.1+ · Vanilla JS

Backend: FastAPI · aiogram 3 · Pydantic 2 · SQLAlchemy 2 (async) · asyncpg · Alembic · Redis · httpx · Clean Architecture / DDD

Устойчивость и данные: Circuit Breaker · Tenacity · rate limiting · restic · SQLite · PostgreSQL

Мониторинг: Prometheus · Grafana · Loki · VictoriaMetrics · Vector · JSON-логи

DevOps / platform: Docker · Compose · Traefik · Let's Encrypt · systemd · GitHub Actions · Poetry · uv · hatchling

Качество: mypy strict · Ruff · pytest · ShellCheck · pre-commit · gitleaks · pip-audit · PyPI Trusted Publishing

Frontend (прикладной): Jinja2 · Tailwind CSS

Опыт

Backend / Platform Engineer — Wednesday Bot

2025.12 – н.в. · Telegram-бот генерации изображений · production v7.4.0 · solo

<https://t.me/wednesdaymorningbot> · <https://github.com/freemspwnz/wednesday>

Асинхронный backend: слоистая архитектура, устойчивость к внешним API и мониторинг в production.

- Многослойный backend (~400 модулей): домен отдельно от ORM и HTTP, политики доступа.
- Пайплайн генерации к GigaChat: retry → circuit breaker → rate limit; Redis-кэш поверх SQLAlchemy 2 async + PostgreSQL.
- Мониторинг: Prometheus + Grafana / Loki; CI — мypy strict, Ruff, ~110 тестов, multi-stage Docker.

Стек: Python 3.12 · aiogram 3 · SQLAlchemy 2 / PostgreSQL · Redis · httpx · Tenacity · Prometheus · Grafana / Loki · Docker · GitHub Actions

Open Source Maintainer — asyncbreaker

2026.04 – н.в. · asyncio circuit breaker · v2.1.0 на PyPI · solo <https://pypi.org/project/asyncbreaker/> ·

<https://github.com/freemspwnz/asyncbreaker>

Библиотека устойчивости для Python: без runtime-зависимостей, чтобы интеграции останавливались предсказуемо, а не каскадом.

- Спроектировал и опубликовал на PyPI единый async API (`await breaker.call` / декоратор только для coroutine): OPEN-окно и пробный вызов в HALF_OPEN.
- State machine CLOSED → OPEN → HALF_OPEN с хранилищем в памяти или Redis — общее состояние на несколько воркеров.
- Релиз: uv + hatchling, CI на Python 3.10–3.13 (pytest, мypy, Ruff, pip-audit), публикация на PyPI по тегам.

Стек: Python 3.10–3.13 · asyncio · redis.asyncio · pytest · мypy · Ruff · uv / hatchling · GitHub Actions · PyPI Trusted Publishing

Platform / SRE Engineer — backup-utils

2026.02 – н.в. · Encrypted restic-бэкапы · Linux-клиент · solo

<https://github.com/freemspwnz/backup-utils>

Модульный bash-клиент (~1.7k LOC): dumps → backup → forget/prune → check → отчёт в Telegram. Несколько jobs на одном хосте, systemd timer.

- Слои вместо одного скрипта: конфиг, restic, дампы, блокировки, уведомления.
- Два бэкенда репозитория (SFTP и REST+TLS); если дамп упал — бэкап не пишется; без гонок по одному геро.

- Политика хранения и проверка целостности, отчёты в Telegram, ShellCheck в CI, секреты вне git.

Стек: Bash 4.1+ · restic · systemd · OpenSSH/SFTP · SQLite / PostgreSQL · Telegram Bot API · ShellCheck · GitHub Actions

Full-Stack Engineer — wedding (invitation + RSVP)

2026.08 · SSR-лендинг и RSVP API · production за Traefik/TLS · solo

<https://github.com/freemspwnz/wedding>

Сайт-приглашение: контент из YAML без пересборки, ответы гостей в JSON, уведомления в Telegram.

- FastAPI + Jinja2 SSR и REST RSVP на Pydantic 2; контент в `config.yaml`, обновляется без rebuild образа.
- Форма RSVP с проверкой на сервере, запись ответов в volume, статус на странице; сбой Telegram не откатывает сохранение.
- Production: Docker Compose, секреты через `.env`, Traefik + Let's Encrypt.

Стек: Python 3.12 · FastAPI · Pydantic 2 · Jinja2 · Tailwind CSS · Docker / Compose · Traefik · Let's Encrypt

Бизнес-аналитик. Клиентские обращения — Сбербанк

2024.12 – 2025.10 · Москва / hybrid

- Считал FCR на 3+ млн обращений: новая методика (+20% к точности), дашборд сократил разбор с часа до 10 минут.
 - Python (pandas) / SQL / Qlik Sense: отчётность для руководства и постановка задач разработке по метрикам.
-

Бизнес-аналитик. Развитие качества — Т-Банк

2021.02 – 2024.12 · Москва / hybrid

- Автоматизировал разметку ошибок в консультациях (15% → 5%); ТЗ разработке и A/B — SLA по задачам 80% → 95%.
 - Нашёл топ-10 проблем поддержки: CSAT 4.1 → 4.5.
-

Образование

Учебное заведение	НИУ ВШЭ
Специальность / степень	Инфокоммуникационные технологии и системы связи
Годы	2015 – 2019

Как работаю

- Домен отдельно от ORM и HTTP — проще тестировать политики и менять инфраструктуру.
- Внешние вызовы: `retry` + `circuit breaker` + `rate limit`, с метриками и понятной остановкой, а не «тихим» каскадом.
- Метрики, логи и алерты живут вместе с сервисом, не «добавим потом».
- Docker, CI, секреты вне `git`, воспроизводимый релиз. Solo/MVP довожу до состояния, в котором можно отдавать в эксплуатацию.

Дополнительно

Языки	Русский — родной · English — B2 / Technical
Open Source	<code>asyncbreaker</code> (PyPI)
Интересы в работе	Platform engineering, reliability, developer experience, async Python