

# Дмитрий Лунев

89188739392 | dima.lunev14@gmail.com

## Обо мне

Backend Python разработчик с более чем 4-летним коммерческим опытом работы в различных сферах, таких как B2B, здравоохранение, ветеринария, HR и другие. Имею 2 года опыта руководства сотрудниками. Работал в командах от 4 до 30 членов, как удаленно, так и в офисе. Участвовал во всех этапах полного цикла разработки: от сбора требований заказчика до создания полного автоматизированного процесса сборки и релизов (CI/CD).

Помимо основной деятельности Backend-разработчика, я также занимаюсь наставничеством и повышением компетенций других разработчиков. Можно сказать, что это мое хобби.

## Коммерческий опыт

Compel  
Программист бэкэнд | 08/2024 - Present

Проект: B2B микросервисы

- Повысил покрытие тестами с 27% до 95% по всем микросервисам, что значительно улучшило надежность и выявляемость ошибок
- Оптимизировал импорт электронных компонентов, сократив время блокировки сервиса с 6 часов до 20 минут, ускорив обновление базы в 18 раз
- Ускорил поиск электронных компонентов, снизив среднее время отклика сервера с 1,2 секунды до 112 мс (в 10 раз), повысив скорость обработки запросов
- Спроектировал архитектуру микросервисов и API с акцентом на повторное использование, снизив дублирование кода и упростив масштабирование
- Оптимизировал процессы тестирования и системной аналитики, улучшив качество релизов и снижая затраты на устранение ошибок
- Интегрировал Kafka с FastStream, исключив блокировку клиента при заказе печатных плат и обеспечив бесперебойную обработку запросов.
- Участвовал в проектировании микросервисной архитектуры, формируя требования к новым сервисам и разделяя монолитные решения на независимые модули.

Стек: Python, Pydantic, SQLAlchemy, PostgreSQL, FastAPI, MinIO, PyTest, Kafka, Kibana, Sentry, KeyCloak, Behave, FastStream, Grafana.

Проект: Сервис для performance review сотрудников компании

- Оптимизировал агрегацию оценок сотрудников, снизив среднее время отклика сервера с 5 секунд до 800 мс, что ускорило генерацию отчетов в 6 раз
- Реализовал пагинацию в отчете о сотрудниках отдела, улучшив удобство использования и снизив нагрузку на сервер при загрузке данных
- Выполнил рефакторинг сервиса, приведя его к единому стандарту с остальными сервисами, что упростило поддержку и масштабирование

Стек: Python, Aiohttp, rypika, PostgreSQL, minio, keycloak, pytest.

Fortech  
Руководитель Python Направления | 06/2022 - 08/2024

Обязанности и основной род деятельности:

- Менторство и наставничество: обучение новых сотрудников, поддержка их адаптации, разбор технических вопросов и помощь в освоении стека
- Оценка кандидатов: проведение технических интервью, скринингов и аттестаций разработчиков (Python)
- Анализ и улучшение процессов найма: участие в разработке и оптимизации системы оценки технических навыков кандидатов

Проект: Фитнес-приложение, позволяющее вести учет потребляемых калорий, планировать питание и отслеживать спортивные активности

- Интегрировал API продуктов, автоматизировав расчет пищевой ценности на основе состава
- Разработал систему подсчета калорий с гибкими единицами измерения (граммы, миллилитры, порции и т. д.)
- Реализовал журнал учета КБЖУ и питания, позволив пользователям отслеживать баланс нутриентов
- Имплементировал сервис для работы с фото и видео (Django + MinIO), что позволило загружать и хранить изображения блюд и видео с тренировками
- Добавил раздел статей и спортивных активностей с видеоуроками, улучшив пользовательский опыт

Стек: Django, Postgresql, Redis, RabbitMQ, GitHub actions, MinIO

Проект: Desktopное приложение для автоматического распознавания номеров автомобилей и аудита веса машин на въезде и выезде

- Спроектировал архитектуру и интерфейс приложения, обеспечив удобство использования
- Имплементировал распознавание номеров на фото и видео с помощью Tesseract OCR, повысив точность идентификации
- Настроил установщик Windows-приложения с помощью PyInstaller, упростив развертывание у заказчиков
- Реализовал систему динамических настроек через ini-файл, что позволило гибко изменять параметры без повторной сборки приложения
- Интегрировал API весов, автоматизировав контроль веса автомобилей и повысив точность учета

Стек: Python3, OpenCV, PyQt6, SQLAlchemy, pyinstaller, SQLite, Docker, GitHub, Trello

Lukoil  
Backend-разработчик | 11/2021 - 06/2022

Проект: сервис кадрового электронного документооборота

- Реализовал генерацию документов по docx-шаблонам, что позволило сократить ручной труд HR-специалистов
- Внедрил систему ролей и прав доступа, обеспечив гибкое разграничение доступа к документам и повысив уровень безопасности
- Добавил возможность электронной подписи документов (ПЭП и УНЭП), что увеличило скорость оформления документов
- Интегрировал сервис с системой уведомлений через Kafka
- Оптимизировал работу базы данных, устранил блокировки и инциденты высокой загрузки
- Разработал и внедрил план разделения сервиса на микросервисную архитектуру, что повысило отказоустойчивость и упростило масштабирование

Стек: Python, Pydantic, SQLAlchemy, PostgreSQL, Redis, FastAPI, MinIO, Portainer, PyTest, Kafka

Проект: онлайн-приём кандидатов на работу

- Реализовал гибкий конечный автомат смены статусов кандидата, обеспечив возможность кастомизации процесса под потребности бизнеса
- Интегрировал сервис с системами верификации паспортных данных и парсинга документов
- Реализовал интеграцию с кадровым документооборотом, обеспечив автоматическое создание и подписание документов при найме
- Интегрировал сервис с системой уведомлений через Kafka
- Добавил систему динамических дополнительных полей и блоков, увеличив гибкость формы анкеты без внесения изменений в код
- Внедрил верификацию кандидатов через почту и телефон, повысив достоверность данных и снизив уровень фрод-активности

Стек: Python, Pydantic, SQLAlchemy, PostgreSQL, Redis, FastAPI, MinIO, Portainer, PyTest, Kafka

Mobicult  
backend-разработчик | 06/2020 - 11/2021

Проект: Мобильное приложение для владельцев питомцев

- Интегрировал API ветеринарных клиник (VetManagerAPI), обеспечив автоматический доступ к записям
- Реализовал интеграцию с платежной системой CloudPayments для оплаты первичного приёма, а также пожертвований в приюты
- Взаимодействовал с заказчиком, формализовал бизнес-требования
- Имплементировал систему напоминаний, обеспечив пользователям возможность автоматического планирования ухода за питомцем.
- Разработал модерацию новых клиник, что ускорило процесс проверки и публикации
- Создал систему кастомизации онбординга через админ-панель, повысив конверсию новых пользователей
- Внедрил авторизацию через подтверждение номера телефона
- Реализовал поиск мероприятий и ключевых точек на карте (PostGIS, GeoAlchemy2), обеспечив удобную навигацию по мероприятиям и ключевым точкам для владельцев питомцев

Стек: Python, Pydantic, SQLAlchemy, PostgreSQL, FastAPI, PyTest, CloudPayments, PostGIS, GeoAlchemy2, Celery, RabbitMQ, Redis

Проект: админ-панель для отправки заявок на отгрузку паллет

- Переписал взаимодействие с API внешнего сервиса с XML на JSON
- Добавил очередь заявок для предотвращения блокировок
- Реализовал журнал событий админ-панели, обеспечив полный аудит действий пользователей и повысив прозрачность работы системы

Стек: Python, sanic, aiohttp (client), PyTest, PostgreSQL, RabbitMQ, Celery

Проект: приложение для сети частных медицинских клиник

- Оптимизировал алгоритм синхронизации данных, сократив время обновления записей
- Добавил кэширование данных из API клиник, что снизило нагрузку на сервер и ускорило загрузку данных для пользователей
- Осуществлял поддержку и доработку приложения, повышая стабильность работы сервиса

Стек: Python, aiohttp (client + server), PyTest, PostgreSQL, Celery, RabbitMQ, Redis

Навыки и Технологии

Python, Django, DRF, FastAPI, Flask, SQLAlchemy, OpenCV, PostgreSQL, Redis, RabbitMQ, Kafka, Git, Docker, Nginx, Kubernetes, AWS, MinIO

Образование

ЮРГПУ (НПИ) имени М.И. Платова  
МОАИС | 05/2022  
Факультет информационных технологий и управления

Кафедра программного обеспечения вычислительной техники

Направление математического обеспечения и администрирования информационных систем

Языки

Английский, Русский