

Коноваленко Кирилл

+7-924-102-98-90 | Physeex@gmail.com | TG: [@HaenesS](#) | GitHub: [Haenes](#) | [Портфолио](#)

НАВЫКИ

BACKEND РАЗРАБОТКА | Python, FastAPI, Django, DRF, Flask, Pytest, Unittest, Unicorn, Nginx

FRONTEND РАЗРАБОТКА | HTML, CSS, JavaScript, React, react-router, Tailwind CSS, Bootstrap

БАЗЫ ДАННЫХ | PostgreSQL, MySQL, Redis

РАЗНОЕ | Git, SQLAlchemy, Alembic, Celery, RabbitMQ, Aiogram, aiogram-dialog, Docker, Ubuntu

ПРОЕКТЫ

BugTracker

- Создал с нуля веб-приложение для отслеживания ошибок с изначальным стеком: Django, HTML, Bootstrap и JS. В котором можно создавать разные проекты и задачи с ними связанные. Управлять ими, отслеживать прогресс, изменять цветовой режим/язык сайта
- Реализовал регистрацию и сброс пароля с подтверждением по электронной почте
- Внедрил Redis для кэширования запросов к базе данных и сложных HTML-шаблонов с аннулированием кэша через сигналы Django
- Разработал функционал Drag'n'Drop для задач с использованием JS и Fetch API
- Позже, полностью переписал его как SPA используя новый стек: FastAPI, React, react-router and Tailwind CSS. С прежним функционалом, добавив полную асинхронность, JWT и Bearer token аутентификацию, для браузеров и API, соответственно, внеся изменения в базу данных и добавив Alembic для отслеживания миграций.

BugTracker REST API

- Создал REST API изначально используя Django REST Framework, который позволил создать telegram-бота на основе данных из первого проекта. Став связующим звеном между ними
- Внедрил Redis для кэширования запросов к базе данных
- Реализовал Token based аутентификацию
- Реализовал Throttling для ограничения частоты запросов к API
- Позже, полностью переписал используя FastAPI, что дало асинхронность. Внедрил JWT и Bearer token аутентификацию. Обратный прокси-сервер ответственен за троттлинг.

BugTracker Telegram Bot

- Создал асинхронного Telegram-бота используя aiogram, оставив весь базовый функционал веб-версии, но уже в виде бота
- Разработал систему входа, которая получает токен пользователя на основе его учетных данных для дальнейшего использования ботом
- Внедрил базу данных PostgreSQL через SQLAlchemy async ORM для хранения информации
- Внедрил Redis для кэширования запросов к БД и в качестве постоянного хранилища FSM
- Реализовал возможность смены языка и часового пояса
- Разработал pagination для проектов и задач, что улучшило удобство использования бота
- Позже, полностью переписал его используя библиотеку aiogram-dialog, что вкупе с новым асинхронным API, значительно улучшило пользовательский опыт. Также я добавил Alembic для отслеживания миграций базы данных.