

Микросервисное приложение для подсчета калорий и трекинга тренировок

Цель данного приложения — предоставить пользователю удобный инструмент для контроля над рационом питания и физической активностью. Архитектура построена по микросервисной модели, что обеспечивает модульность, масштабируемость, отказоустойчивость и независимое развитие компонентов. Пользователь получает доступ к функционалу через фронтенд-интерфейс (мобильный или веб), взаимодействующий с API Gateway, который маршрутизирует запросы по соответствующим микросервисам.

Frontend

Фронтенд — это пользовательский интерфейс, реализованный с использованием современных JavaScript-фреймворков (например, React или Vue). Он отвечает за: отображение интерфейса; сбор пользовательского ввода; отправку запросов через API Gateway; обновление данных в реальном времени.

API Gateway

Центральная точка входа всех HTTP-запросов. Отвечает за: аутентификацию (проверку JWT токенов); ограничение частоты запросов; логирование; маршрутизацию запросов к соответствующим микросервисам; обратную прокси-функциональность.

Backend

Центральный управляющий компонент, который может содержать бизнес-логику, координировать работу сервисов и выполнять сложные операции. Также может агрегировать ответы от нескольких сервисов.

Сервис Пользователей

Занимается регистрацией, авторизацией и хранением пользовательских данных. Реализует: CRUD-операции над профилем пользователя; настройки целей по калориям и тренировкам; обработку JWT токенов; связь с БД для хранения информации.

Сервис Питания

Регистрирует данные о приёмах пищи, подсчитывает калории, макроэлементы (белки, жиры, углеводы). Включает: базу продуктов с энергетической ценностью;

логирование приёмов пищи; расчет дневной суммы потребленных калорий; возможность создавать собственные рецепты и блюда.

Сервис Тренировок

Отвечает за сбор и хранение информации о выполненных пользователем физических упражнениях. Его функциональность: учет тренировок по типу, длительности и интенсивности; расчет сожжённых калорий; планирование тренировок; возможность интеграции с фитнес-трекерами.

Сервис Аналитики

Агрегирует данные из сервисов питания и тренировок для формирования статистики, прогресса и рекомендаций. Основные задачи: расчет ежедневного баланса калорий; построение графиков и трендов; выявление отклонений от цели; подготовка отчетов для фронтенда и уведомлений.

Сервис Уведомлений

Отправляет пользователям сообщения в зависимости от событий (новый день, достижения, отклонения). Использует очереди сообщений для: email, push, SMS-уведомлений; напоминаний о тренировках; персонализированных рекомендаций на основе аналитики.

Брокер (Очередь сообщений)

Связывает сервисы событийным способом. Используется, чтобы: обеспечить асинхронное взаимодействие между сервисами; разгрузить основную систему при больших объемах событий; повысить отказоустойчивость (сообщения не теряются при сбоях).

Базы данных

Каждый микросервис работает с собственной БД. Микросервисная архитектура предполагает использование разделенных баз данных для каждого сервиса, что позволяет избежать лишних зависимостей и облегчает масштабирование отдельных компонентов.

