

Микросервисная архитектура для осуществления онлайн-покупок

1. Frontend (React/Vue.js + TypeScript)

Frontend отвечает за взаимодействие с пользователем и визуализацию данных, общается исключительно через API Gateway, путем отправки REST-запросов. Основные функции:

- Оформление заказов
- Просмотр товаров
- Управление профилем
- Отображение статуса заказов

2. Backend (API Gateway + микросервисы)

Backend состоит из нескольких независимых сервисов, взаимодействующих через REST API, gRPC или асинхронные сообщения (Kafka/RabbitMQ).

2.1. API Gateway (Nginx / Kong / Spring Cloud Gateway)

- Единая точка входа для всех запросов от Frontend-части
- Маршрутизация запросов к соответствующим сервисам
- Аутентификация и авторизация (JWT/OAuth2)
- Логирование, кеширование, ограничение запросов (Rate Limiting)

3. User service (сервис пользователя)

Сервис пользователей выполняет следующие функции:

- Регистрация, аутентификация (JWT/OAuth2)
- Управление профилями (личные данные, адреса доставки)
- Роли и права доступа (покупатель, администратор)
- Восстановление пароля, верификация электронной почты

4. Product catalog service (сервис каталога товаров)

Сервис каталога товаров позволяет пользователям просматривать товары, применять фильтры (цена, категория, рейтинг) и искать нужные позиции через поисковую систему. Сервис предоставляет детальную информацию о товарах

(фотографии и описание, цена и наличие на складе, отзывы и рейтинги). Также сервис может формировать персонализированные рекомендации на основе истории просмотров и покупок пользователя.

5. Shopping cart service (сервис корзины)

Позволяет пользователю добавлять товары в корзину, изменять их количество и сохранять выбранные позиции для будущих покупок.

Основные функции:

- Добавление/удаление товаров
- Автоматический пересчет стоимости с учетом акций и скидок
- Сохранение корзины между сессиями для авторизованных пользователей

При оформлении заказа корзина передается в Order Service, а затем очищается.

6. Order service (сервис заказов)

Регистрирует и отслеживает заказы пользователей.

Функционал:

- Оформление заказа на основе данных из корзины
- Выбор способа доставки
- Отслеживание статуса заказа
- История заказов и возможность повторного заказа

Интеграции: **Payment service, Inventory service, Notification service**

7. Payment service (сервис платежей)

- Обработка платежей через различные способы (банковские карты, электронные кошельки, криптовалюты (при наличии))
- Возврат средств
- Уведомление Order Service об успешной оплате

8. Склад и доставка

Управляет количеством товаров и логистикой.

- **Inventory service** осуществляет контроль наличия, резервирование товаров при заказе.

- **Delivery service** рассчитывает сроки и стоимости доставки (интеграция с API транспортных компаний)

Дополнительные функции:

- Уведомление о поступлении товара (если была пометка “нет в наличии”)
- Отслеживание посылок (трекинг-номер)

9. Аналитика и рекомендации

Analytic service собирает данные о поведении пользователей и формирует отчеты (популярные товары и тренды, эффективность маркетинговых акций).

На основе этих данных **Recommendation service** предлагает персонализированные рекомендации и скидки для постоянных клиентов.

10. Notification service (сервис уведомлений)

Отправляет пользователям информацию о статусе заказа, акциях и скидках, доставке.

Интеграции: Email-сервисы, SMS-шлюзы, WebSocket для мгновенных уведомлений в браузере.

Итоговая схема работы

1. Пользователь выбирает товары в каталоге товаров и добавляет в корзину.
2. Оформляет заказ через сервис заказов и оплачивает через сервис оплаты.
3. Сервис склада резервирует товары, сервис доставки отправляет заказ.
4. Сервис уведомлений информирует пользователя на всех этапах.
5. Сервис аналитики собирает данные для улучшения сервиса.

