



Описание взаимодействий

- 1-4. **Запросы авторизации** — Пользователь запрашивает токен или проверяет валидность текущего токена. Запрос уходит последовательно на API Gateway и обратно.
5. **Запросы по заказу/аукциону от пользователя** — Пользователь в личном кабинете/портале выполняет запрос по управлению заказом; запросы проходят через API Gateway к сервису управления заказами.
6. **Запросы по работе с аукционом** — Аналогично сервису заказов, работа с сервисом аукциона через API Gateway.
7. **Запросы к платежной системе от сервисов Заказов и Аукционов** — сервисы аукциона и заказа принимают вызовы пользователей через API на создание определенных сущностей (заказ/аукцион), обрабатывают их и вызывают сервис платежей для проведения транзакции. Если говорить о реализации, то я бы все это добавил через событийную систему и реализовал паттерн Saga, для случаев сбоев/отмены платежей. Сервис заказов/аукционов размещает сообщение, например в Kafka и каждый из сервисов подписывается на событие и обрабатывает его.
8. **Обработка платежа** — Сервис платежей делает различные вызовы и предоставляет определенное API для работы с платежами, проведение/отмена/запрос статуса (не стал рисовать для каждой операции свои стрелочки, они все не поместятся).
9. **Мониторинг и поддержка** — Сервисы платежей/аукционов/заказов публикуют статусы (результаты выполнения операции) и сотрудники техподдержки могут наблюдать через средства мониторинга и в случае необходимости предпринимать действия по исправлению ошибок, так же этот сервис имеет отдельную БД для хранения исторических данных по всем операциям пользователей, что так же снижает нагрузку на БД сервисов.

Дополнительные пояснения

Добавлены дополнительные микросервисы, в БД описаны хранимые сущности, в БД заказов, аукционов, платежей например присутствует так же ID пользователя, т.к. эти сущности не могут существовать сами по себе, в БД товаров и услуг в этом нет необходимости.

В данной схеме назначение API Gateway следующее: оно служит единой точкой входа для запросов пользователей, маршрутизирует запросы на соответствующие микросервисы, а так же запросы от самих сервисов друг к другу (например сервис заказов обращается к сервису товаров для формирования заказа).

Помогает улучшить безопасность, например выполняется проверка срока действия токена пользователя, его прав доступа, так же выполняет функции балансировщика (в случае подключения дополнительных реплик сервисов, нагрузка будет распределена между ними).