

Сервис будет обеспечивать Consistency для записи и Partition tolerance.

Сервис состоит из агрегатора и сервисов-приложений.

- Если запрос на чтение, агрегатор опрашивает сервисы самокатов и возвращает ответ - список доступных самокатов.
- При запросе бронирования - агрегатор шлет запрос только одному сервису и возвращает успех/не успех операции.

Для запросов /scooters (чтение) будем пытаться поддерживать availability. Для запросов /scooters/{id}/reservations (создание/удаление) - consistency.

Агрегатор: бд - mongoDB. tables: clients

Агрегатор будет хранить только uuid как маркер того, что пользователь зарегистрирован в системе.

Сервисы: бд -mongoDB, tables: scooters

Сервис будет хранить все в одной коллекции (где документ - scooter_id, is_booked, client_uuid, .price, etc) чтобы можно было делать атомарный findAndUpdate

scooters/, scooters/nearest и availability

Для быстрого тестирования мы пробрасываем в сеть локальный порт машины с помощью ngrok. В бесплатной версии он позволяет быть 40 соединениям в минуту. Чтобы наш сервис совсем не упал при нагрузочном тестировании и чтобы можно было проверить как работают другие части системы мы решили кэшировать список самокатов и запрашивать его раз в минуту.

- Предположение 1: пользователю не обязательно возвращать самые свежие данные потому что пока будет совершен запрос на резервацию данные уже могут утратить актуальность.
- Предположение 2: чем меньше запросов к сервисам тем меньше вероятность упасть нашему сервису.
- Проблема 1: повышается вероятность двойного бронирования.
- Решени: поддержка consistency у сервисов проката при запросе на обновление.

запросы **/scooters/id/reservation**

- в агрегаторе чтение uuid. Для обеспечения CP используется mongoDB с replica set с 3мя нодами и стратегией read-write - "majority". То есть если uuid был создан-записан в бд, то агрегатор не ругнется что используемый uuid неверный.
- Запрос к сервису для создания reservation. так как нам важно чтобы не произошла повторная бронь уже забронированного самоката, будем обеспечивать Consistency, используя mongoDB с настройками: replica set, 3 ноды, стратегия read/write - "majority" + preference: read from Primary и findAndUpdate(который если верить мануалу является атомарным).