**Тестовое задание**

**Используемые технологии (** Пожалуйста не используйте то что не указано в этом списке **)**

TypeScript

Node.js ( Можно использовать Express или Fastify )

MongoDB ( Без ORM )

WebSockets ( Можно использовать только https://github.com/websockets/ws )

**Описание**

Вам необходимо создать сервер который будет работать с сообщениями. Не требуется хостинг, только предоставление кода.

Сервер на Node.js должен иметь следующие http роуты:

• POST Создать сообщение

• GET Получить все сообщения

Сообщения должны храниться в MongoDB.

При создании сообщения сервер должен накапливать их пачками по 10 штук и отправлять в MongoDB всю пачку.

Накопление должно происходить по принципу что быстрее произойдет - накопится 10 сообщений или пройдет таймаут в 1 секунду.

При отключении сервера сообщения не должны быть потеряны.

Сервер должен “слушать” изменения в MongoDB и отправлять отдельное WebSocket событие о каждом добавленном в БД сообщении.

При включении сервера он должен продолжить “слушать” и отправлять WebSocket события с того места где остановился, не пропустив не одного изменения в БД.

**Оцениваемые аспекты**

• Дизайн и структура системы

• Накопление и обновление данных

• Все требования соблюдены и нет багов

• Код продакшн уровня