

# Тестовое задание

## Используемые технологии

Node.js

React

React Query

WebSockets (используем <https://github.com/websockets/ws> для Node.js)

## Описание

Вам необходимо создать простое приложение для обмена сообщениями между клиентом и сервером. Не требуется хостинг, только предоставление кода.

Сервер на Node.js должен иметь следующие http роуты:

- POST Создать сообщение
- GET Получить все сообщения

Созданные сообщения должны храниться в памяти сервера. Объем памяти должен быть ограничен до 9 сообщений. При достижении предела, новые сообщения должны вытеснять старые. WebSocket уведомления должны быть отправлены о добавленных или удаленных сообщениях из памяти.

React-приложение должно использовать библиотеку React Query для взаимодействия с данными сервера. События WebSocket должны обновлять данные в React Query. UI должен отображать актуальный список сообщений и предоставлять инпут для создания новых сообщений.

При запуске клиентское приложение должно получить список сообщений через GET запрос к серверу, затем обновления списка сообщений должны приходить через WebSocket.

## Оцениваемые аспекты

- Дизайн и структура системы
- Взаимодействие между клиентом и сервером через WebSockets
- Управление состоянием и обновление данных
- Логика контроля лимита памяти

## Что не будет оцениваться

- Стили пользовательского интерфейса (UI)

Дедлайн – 2 дня (если не успеваете доделать - скидываем то, что готово), результат принимаем ссылкой на github