Команда: Коливо ~ don’t ask, why. Just understand, because…

Кейс: Back-end.

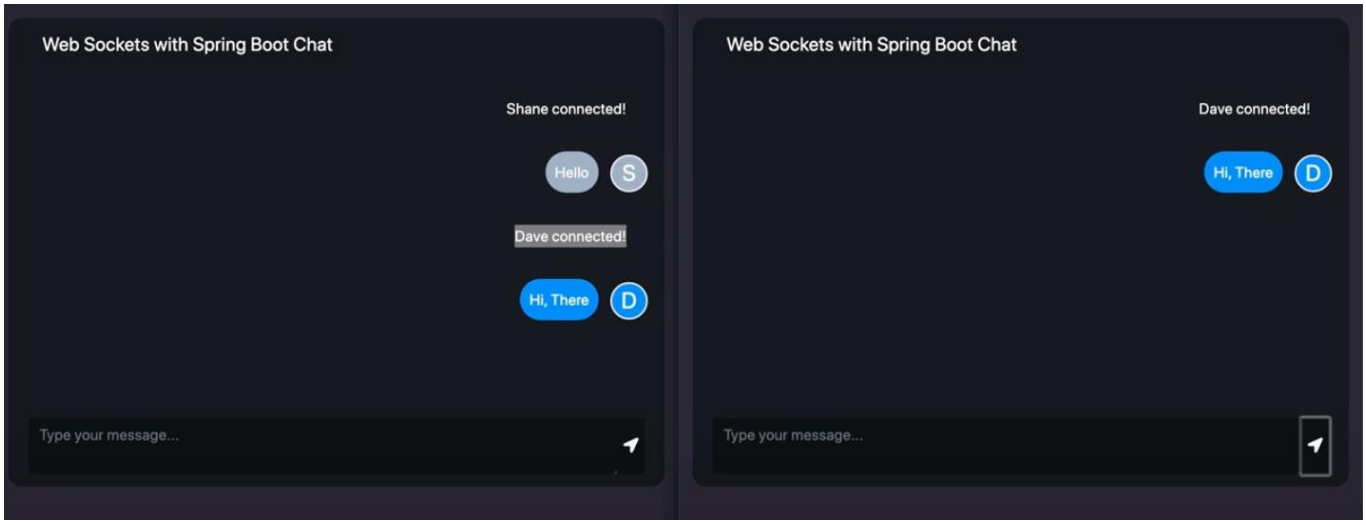
Использованные технологии: Spring Framework удобен, так как с использованием DI легче реализовать необходимый функционал, также благодаря связным классам дальнейшая разработка становится проще, Spring Boot – для обмена сообщениями использовались WebSockets, Spring Data, Spring Security, PostgreSQL, JWT, сборка maven, для реализации REST API были использованы RestControlles и XMLHttpRequest

КАК БЫЛО ЗАДУМАНО: пользователь авторизируется (регистрация/вход), далее попадает в свой аккаунт. В аккаунте список чатов, в которых участвует пользователь, количество человек – от 2 до 100. Можно выбрать случайный чат для общения с незнакомыми людьми. Все чаты и данные пользователей хранятся в таблицах БД, при открытии чат подгружается в сессию.

Как дополнительная фича – отдельные открытые чаты: такой вот клаб-хаус, но с буквами. Пользователь вводит тему, на которую ему интересно поговорить, например, “java”, в поиске по времени выдаются сообщения, содержащие это слово. Пользователь может перейти в чат и присоединиться, если ему станет интересно.

Что мы успели сделать: Работающий чат (случайный и обычный чат + индивидуальные сообщения), основанный на WebSockets и контроллерах, даже с небольшим фронт-эндом (чтобы удобнее было тестить, они не связаны в конечной версии). Авторизацию с использованием jwt, добавление пользователей в БД и поиск по БД. Сохранение чатов в БД, связь таблиц пользователей и чатов (ManyToMany), поиск чатов по названию to be continued…

Что-то вроде примера (да, мы помним, что в задании бэк):



Спасибо за внимание!