

Тестовое задание backend

Задача

Разработать REST API сервис для онлайн чата (только backend). Предположим, что данный REST API сервис будет использоваться абстрактным мобильным приложением.

Общее описание возможностей сервиса

Можно зарегистрироваться и вести личную переписку с любым другим пользователем проекта. Переписку не в своих чатах увидеть нельзя.

Список методов (возможностей) REST API

- Регистрация пользователя. Передаваемые поля: email, password, lastName, firstName.
- Получение accessToken через проверку email и password.
- Получение списка пользователей сервиса. Возвращаемые поля: userId, email, lastName, firstName. Список возвращать порционно по 20 пользователей за один запрос.
- Создать чат с любым пользователем проекта (если еще не создан) - другими словами возможность пригласить пользователя в чат, передав его userId.
- Отправить текстовое сообщение в конкретный чат по chatId.
- Получить список чатов, где текущий пользователь является участником (чужие чаты пользователь не должен увидеть). Сортировать по timestamp последнего сообщения в чате по убыванию (чтобы свежие чаты были вверху списка). Возвращаемые поля: chatId, timestamp, список участников, название чата (строить исходя из того, с каким пользователем ведется переписка, например: Иванов Иван). Список возвращать порционно по 20 чатов за один запрос.
- Получить список сообщений для конкретного чата (сообщения чужих чатов пользователь не должен увидеть). Более свежие сообщения должны быть вверху списка. Возвращаемые поля: messageId, timestamp, text, кто отправитель сообщения. Список возвращать порционно по 20 сообщений за один запрос.

Стек технологий

Laravel 11

Laravel Sail

Laravel Sanctum

Scribe <https://github.com/knuckleswtf/scribe/>

MySQL

Docker

Обязательные требования

- В качестве БД использовать MySQL.
- Код должен быть оформлен по стандартам PSR-1 и PSR-2.
- Все запросы, кроме регистрации пользователя и получения токена, должны быть защищены и требовать аутентификацию через передачу access токена. Для выдачи access токена использовать Laravel Sanctum.
- Результаты REST API методов возвращать в формате json.
- При проектировании REST API методов придерживаться правил именования ресурсов <https://restapitutorial.ru/lessons/restfulresourcenaming/>
- Проект должен разворачиваться и работать через docker-compose. Для этого необходимо использовать Laravel Sail. В файле [README.md](#) необходимо описать точные шаги, которые позволят поднять проект на сервере при помощи docker-compose.
- Для всех REST API методов должна быть сгенерирована документация. В качестве движка документации использовать Scribe <https://github.com/knuckleswtf/scribe/>
- Для всех REST API методов нужно написать простые Feature автотесты, которые будут проверять работает ли конкретный REST API метод (возвращается ли код 200). Unit тесты писать не нужно.
- Реализовать в REST API методах возможность версионирования. Чтобы в случае кардинальных изменений в REST API методах не ломать обратную совместимость для клиентов.
- В сервисе должен быть реализован RateLimit с ограничением 10 RPS с одного IP.
- Оформить логичные коммиты в GIT (весь проект одним коммитом не подойдет).
- Код выложить на github.