

Тестовое задание для Backend Developer C#

Версия 3 от 1 июня 2022

<https://irkutsk.hh.ru/vacancy/55450426>

Если нашли ошибку или у вас есть вопросы или предложения, то напишите нам на почту job@megame-studio.com

Время для выполнения 2 недели.

Вступление

Требуется разработать систему поддержки для игр.

Игроки пишут сообщения в поддержку прямо из игры, а операторы поддержки отвечают на сообщения игроков.

Система состоит из следующих частей:

- административный интерфейс - далее админка - тут операторы читают и отвечают на сообщения игроков;
- апи для обработки запросов из игры;
- клиентская библиотека, использующая апи для общения с сервером из игры;

Каждое задание ниже, это часть одной целой системы.

Детали, не указанные в описании задания, вы можете придумать самостоятельно так как удобно вам или считаете нужным. Вы можете указать это в комментарии к выполненному заданию.

Задание 1 - Админка

Сделайте админку для операторов, со следующими страницами:

Авторизация - операторов может войти в систему при помощи email и пароля.

Список диалогов - оператор может видеть список диалогов от игроков.

Страница диалога - оператор может видеть диалог с игроком..

Форма ответа на странице диалога - оператор может написать новое сообщение.

Структуры данных

Игрок

- Токен устройства - строка, уникальна для каждого игрока, используется для авторизации.
- Никнейм - строка, максимум 16 символов.

Оператор

- Почта
- Пароль
- Имя

Сообщение

- Игрок - для сообщений от игрока
- Оператор - для сообщений от оператора
- Время создания
- Прочитано / не прочитано
- Текст сообщения

Задание 2 - Http api

Создайте следующие методы api

- Регистрация игрока - добавляет нового игрока в систему;
- Кол-во непрочитанных сообщений;

Задание 3 - WebSocket api

- Авторизация игрока по Токену устройства;
- Получить список сообщений с пагинацией и возможностью запросить следующую "страницу";
- Отправить сообщение;
- Отметить сообщение как прочитанное;

Задание 4 - Роли

Добавьте разграничение доступа для пользователей админки:

- Гость - может только войти в систему;
- Наблюдатель - может только читать;
- Оператор - может и читать и писать;

Задание 5 - Документация

Сгенерируйте документацию для api любым удобным для вас способом.

Задание 6 - Кэширование

Добавьте кэширование на стороне сервера для запроса:

- Кол-во непрочитанных сообщений;

Используйте redis.

Задание 7 - Клиент на Unity

Напишите клиент для [Unity](#), который реализует апи, созданное в заданиях 2 и 3.

Для этого задания пользовательский интерфейс не обязателен, достаточно, чтобы ответы сервера выводились в консоль, а код выполнялся в среде Unity.

Рекомендуем использовать библиотеку Best HTTP для отправки запросов на сервер.

[Документация](#)

[Скачать](#)

Требования

- C#
- .NET Core
- Формат данных для апи = json
- В качестве базы данных используйте MySQL или любую другую реляционную
- WebSocket
- Redis для задания с кешированием

Можно использовать библиотеки для веб сокетов, сериализации, подключения к mysql, redis, Entity Framework и другие.

Нельзя использовать готовые ассеты и чужой код, решающий в значительной степени задачу, универсальные большие шаблоны верстки (bootstrap и подобные разрешается).

Нельзя использовать сторонние сервисы в качестве зависимостей - aws, yandex cloud и другие.

Мы ожидаем, что вы в состоянии самостоятельно написать всю бизнес логику для решения задачи.

Если у вас вопрос по использованию какого либо готового компонента, пожалуйста, напишите нам job@megame-studio.com

Результат

- Запишите небольшое видео с результатом работы (1-5 минут). Выложите на youtube (установите доступ по ссылке). Перед записью удостоверьтесь, что на видео видно курсор мыши.
- Загрузить исходный код проекта на github в приватный репозиторий, дайте доступ для пользователя по email: dimmduh@gmail.com

Оцениваться будет:

- Соответствие заданию;
- Техническое решение;

Оцениваться НЕ будет:

- визуальная составляющая

Как отправить результат

Отправьте ссылками на почту

job@megame-studio.com

P.S. В своей работе мы ваше тестовое задание использовать не будем.

Часто задаваемые вопросы по тестовому заданию

Сколько у меня времени на выполнение тестового задания?

2 недели с момента получения задания.

Почему нужно загружать в приватный репозиторий?

Тестовое задание выполняют несколько кандидатов, и чтобы исключить возможность “списывания” друг у друга, необходимо закрыть публичный доступ.


В задании есть спорные моменты и требования?

Да, мы понимаем, что в заданиях иногда требуется не совсем “правильная” реализация. Например, в задачи про кэширование можно было бы выполнить это на стороне nginx (например).

Могу ли я впоследствии использовать результат тестового задания в своём портфолио?

Да.

Часто задаваемые вопросы по работе в компании

 Часто задаваемые вопросы по работе в компании Megame