YONOTE

Тестовое задание на позицию Junior Backend Developer

Используемые инструменты

Для выполнения этого задания необходимо использовать следующие технологии:

- Go,
- PostgreSQL,
- Docker.

Ваше решение должно запускаться одной командой:

docker-compose -f docker-compose.yml up -d.

Запуск должен происходить на основе дефолтных значений, указанных в конфигурационных файлах (включая .env).

Задание

Необходимо реализовать часть сервиса аутентификации. Сервис должен предоставлять четыре конечные точки API:

- на получение пары токенов (access и refresh) для пользователя с идентификатором (GUID) указанным в параметре запроса;
- на обновление пары токенов;
- на получение GUID текущего пользователя (роут должен быть защищен);
- на деавторизацию пользователя (поле выполнения этого запроса с access токеном, пользователю больше не должен быть доступен роут на получение его GUID и операция обновления токенов).

Требования

Требования к access токену:

1. Формат токена - JWT,
2. Алгоритм для подписи токена - SHA512,
3. Хранить токен в базе строго запрещено.

Требования к refresh токену

1. Формат токена - произвольный.
2. Передаваться токен должен только в формате base64 .
3. Хранить токен в базе строго в виде bcrypt хеша.
4. Токен должен быть защищен от повторного использования.
5. Токен должен быть защищен от изменений на стороне клиента.

Требования к операции refresh

1. Операцию refresh можно выполнить только той парой токенов, которая была выдана вместе.
2. Необходимо запретить операцию обновления токенов при изменении User-Agent . При этом, после неудачной попытки выполнения операции, неоходимо деавторизовать пользователя, который попытался выполнить обновление токенов.
3. При попытке обновления токенов с нового IP необходимо отправить POST-запрос на заданный webhook с информацией о попытке входа со стороннего IP. Запрещать операцию в данном случае не нужно.

Результат работы

Результат должен быть представлен в виде исходного кода на GitHub.

Обязательным условием для проверки вашего решения является:

- наличие `docker-compose` файла и возможность развернуть ваш сервис (вместе с БД) одной командой - `docker-compose -f docker-compose.yml up -d` , на основе дефолтных настроек и конфигураций;
- наличие swagger документации, в которой описаны возможные ошибки и есть примеры запросов на каждую реализованную конечную точку API.

Друзья! Задания, выполненные полностью или частично с использованием ChatGPT видно сразу. Если вы не готовы самостоятельно решать это тестовое задание, то пожалуйста, давайте будем ценить время друг друга и даже не будем пытаться :)