

Тестовое задание для Back-End разработчиков.

Задание состоит из двух частей. К выполнению второй можно приступить только после успешного выполнения первой части.

Задание 1. Создать простое webApp.

- Создать простое webApp, используя Node.js (Express, JavaScript), PostgreSQL (Sequelize ORM).
- При запуске приложение должно создать в базе данных таблицу “users” с помощью миграции и добавить в неё один пользовательский аккаунт, на котором будет лишь одно поле “balance” со значением 10000. Для совершения миграций, управляемых приложением, можно использовать библиотеку “Umzug”.
- Написать route для обновления баланса пользователя, как в большую, так и в меньшую сторону, принимающего параметры userId и amount.
- Важным условием является то, что баланс пользователя не может быть отрицательным.
- Изменение баланса должно производиться в реальном времени, без использования очередей и отложенных задач.

Данное задание будет тестироваться отправкой 10000 запросов в один момент на попытку снять с баланса пользователя по 2 единицы. Успешно должно отработать 5000 запросов, вторая их половина должна получить адекватную ошибку о том, что средств на балансе недостаточно.

Инструкция:

1. После завершения работы нужно загрузить её на [github](#) и предоставить ссылку для проверки В ТГ https://t.me/vitaliya_ob.
2. При отправке ответа, пожалуйста, сообщи свои фамилию и имя в сообщении. Наш HR максимально быстро свяжется с тех лидом для проверки. Фидбэк будем стараться дать быстро - мы ценим время, которое ты потратил на работу - наш HR очень заинтересован, чтобы твоему заданию уделили внимание)
3. Если задание будет выполнено успешно, мы попросим тебя выполнить второе тестовое (ниже) - выполняется только после положительного фидбэка по первому заданию.
4. Мы планируем, что на два задания не должно уйти больше 8 часов.

Задание 2. Продолжение разработки webApp.

- Продолжить разработку webApp, на этот раз нашему приложению требуется обрабатывать клиентов на нескольких серверах одновременно, минимум 5 параллельных копий.
- Нужно добавить в приложение cron-service, в котором будет минимум 10 тестовых фоновых задач.
 - Назвать каждую можно как угодно.
- У каждой задачи должна быть настройка её интервала выполнения, вызываемой ею функции, имени задачи.
- Каждая задача должна выполняться не менее 2 минут.
 - Добавить внутрь обработку данных или sleep, чтоб функция задачи завершилась не ранее, чем через 2 минуты после запуска.
- Каждая задача, если её запустила какая-то из копий приложения, не должна запускаться на другой копии, пока где-либо запущена.
- На одной копии приложения может быть запущено сколько угодно задач.
- При одновременном старте всех копий приложения, копии должны распределить между собой задачи равномерно.
- Выполненные задачи должны записываться историей в базу данных.
- Должна быть возможность в реальном времени получить по ручке список всех фоновых задач, с информацией у каждой, на каком из серверов она выполняется и какое время прошло с момента её запуска. В этом списке также должны находиться задачи, которые ожидают выполнения.
- Важным моментом является то, что ни одна копия приложения не должна ничего знать о другой копии, полное отсутствие мастер-сервера.

Инструкция:

1. После завершения работы нужно загрузить её на [github](https://github.com) и предоставить ссылку для проверки В ТГ https://t.me/vitaliya_ob.
2. При отправке ответа, пожалуйста, сообщи свои фамилию и имя в сообщении.