Тестовое задание
Backend Developer

Предлагаем выполнить тестовое задание на позицию Backend Developer в компании Sfxdx (ООО “Интеграл”).

Задание

Реализовать бэкенд биржи с лимитными и маркетными ордерами. Смарт-контракт реализован и доступен.

Предполагается, что тестовое задание будет выполнено с использованием Nest и будет реализовано Rest API либо Json-rpc (вебсокет). Документация (например Swagger), а также TypeScript горячо приветствуется.

Для выполнения можно использовать любые фреймворки и библиотеки.

Смарт-контракт

Смарт контракт верифицирован и доступен в блок эксплорере. Там же можно найти его abi.
Основные методы:

- createOrder - создает заявку.
- cancelOrder - отменяет заявку.
- matchOrders - пытается исполнить ордер полностью или частично на основе списка переданных противоположных ордеров. На неисполненный остаток размещает ордер на оставшуюся сумму. Список противоположных ордеров предоставляет бэкенд (см описание getMatchingOrders ниже).

Остальные публичные методы можно посмотреть в коде.

Бэкенд

Бэкенд агрегирует созданные ордеры и предоставляет доступ к ним. API содержит всего 2 ендпоинта:

- GET /getOrders?tokenA={адрес токена}&tokenB={адрес токена}&user={адрес пользователя}&active={true|false}. Возвращает массив заявок. Параметры опциональны. Без параметров вернет все ордера. С заданным tokenA\tokenB - все заявки где в паре есть tokenA и|или tokenB. С параметром user - все заявки от конкретного пользователя. Параметр active (по умолчанию false) задает выдачу только не закрытых заявок.
- GET /getMatchingOrders?tokenA={адрес токена покупки}&tokenB={адрес токена продажи}&amountA={сумма покупки, если 0 заявка исполнится по рынку}&amountB={сумма продажи}. Возвращает массив идентификаторов заявок, для вызова метода matchOrders в смарт контракте.

Заявки в бэкенд попадают при прослушивании событий смарт-контракта - OrderCreated, OrderMatched, OrderCancelled. Для подписки на события можно использовать Infura.
Хранить заявки можно в любом подходящем хранилище данных.

По событию OrderCreated - заявка должна сохраняться в хранилище.

По событию OrderMatched - состояние сохраненной заявки должно меняться. Как только amount совпадет с amountLeftToFill, заявка считается закрытой, становится не активной и не далее не участвует в сопоставлении.

По событию OrderCancelled - заявка должна считаться закрытой, быть неактивной, и далее не участвовать в сопоставлении.

При перезапуске бэкенд должен проверять события с момента деплоя контракта и приводить хранимые заявки в актуальное состояние.

[1]Секция A, см дизайн макет.
[2]Секция B, см дизайн макет.
[3]Секция C, см дизайн макет.

Тестовое задание
Backend Developer

Предлагаем выполнить тестовое задание на позицию Backend Developer в компании Sfxdx (ООО “Интеграл”).

Фронтенд

Фронтенд предоставляет возможность размещать ордера (в блокчейне) на покупку или продажу любого токена. Предоставляет список заявок пользователя и текущие цены для указанной пары. Взаимодействие с блокчейном осуществляется с помощью библиотеки web3 и кошелька metamask.

Примечания

Полное описание проекта.
Дизайн макет.
Смарт контракт 0x352f8c1f8576183b6c783d3e589abb69ffbebc47.

Аккаунт и ключ (мнемоника) с тестовыми ETH в сети goerli и тестовыми токенами:
0x22961F4EB722B9582E9743a662e6f1c051add4df
owner super essay kangaroo same virtual predict solid ring absurd canoe merit

ВНИМАНИЕ: Лучше создать чистый профиль в браузере, установить metamask туда, импортировать указанный ключ и ни в коем случае не использовать его для личных нужд в мэйннете - ключ доступен многим.