

Тестовое задание

Frontend Developer

Предлагаем выполнить тестовое задание на позицию Frontend Developer в компании Sfxdx (ООО "Интеграл").

Задание

Реализовать фронтенд (dapp) биржи с лимитными и маркетными ордерами. Смарт-контракт и бэкенд реализованы и доступны. В дизайн макете заложены 3 уровня сложности в виде секций А, В и С. Минимальный уровень - это возможность размещения и исполнения заявок - секция А.

Предполагается, что тестовое задание будет выполнено с использованием React. Для выполнения можно использовать любые фреймворки и библиотеки.

Краткое описание

Приложение предназначено для купли и продажи токенов. Пользователи могут размещать ордера на покупку (продажу) токенов по рыночной или установленной цене¹. Также каждому доступны для просмотра собственные ордера² и текущие зарегистрированные ордера в виде биржевого стакана³.

Основные составляющие - смарт контракт, бэкенд и фронтенд.

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).

Бэкенд агрегирует созданные ордера и предоставляет доступ к ним. API содержит всего 2 эндпоинта:

/getOrders?tokenA={адрес токена}&tokenB={адрес токена}&user={адрес пользователя}

Возвращает массив заявок. Параметры опциональны. Без параметров вернет все ордера.

С заданным tokenA\tokenB - все заявки где в паре есть tokenA или tokenB. С параметром user - все заявки от конкретного пользователя.

/getMatchingOrders?tokenA={адрес токена}&tokenB={адрес токена}&amountA={сумма покупки\продажи}&amountB={сумма покупки\продажи}&isMarket={true|false}

Возвращает массив идентификаторов заявок, для вызова метода matchOrders в смарт контракте.

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 0xC7dd7d4730d95AAE47F27c32eBb85b04fc78769E.](#)

[Бэкенд.](#)

Аккаунт и ключ (мнемоника) с тестовыми ETH в сети rinkeby 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 мейннете - ключ доступен многим.