

Test: Junior FullStack Developer

** Цель задания - проверить навыки работы с React.js/Node.js и с базой данных MongoDB, а также навыки изучения новых технологий*

Задание: Разработать страницу списка транзакций на блокчейне Ethereum

1 React.js

Страница должна состоять из следующих блоков:

1. Блок с фильтрами :
 - Input поиска транзакции (по **адресу получателя/отправителя, id транзакции, номеру блока**), для выбора типа поиска делаем Select с вариантами
 - Кнопка подтверждения поиска
2. Блок списка транзакций, должен быть реализован как таблица с постраничной пагинацией. Сама транзакция должна содержать такую информацию:
 - ID транзакции (ссылка на <https://etherscan.io/>)
 - Адрес отправителя
 - Адрес получателя
 - Номер блока
 - Количество подтверждений (количество блоков которые вышли после блока в котором была транзакция)
 - Дата отправки транзакции (брать с блока)
 - Отправленная сумма
 - Комиссия (<https://ethereum.org/ru/developers/docs/gas/>)
3. Данные по сумме и номеру блока должны выводиться как число

2 Node.js/MongoDb

1. Разработать АПИ которое будет отправлять на Front-end список транзакций по полученным фильтрам (должна быть предусмотрена **валидация данных** полученных в запросе, а также **пагинация**), данные должны браться из базы данных
2. Дополнительно при запуске приложения должна запускаться функция (как вариант рекурсивная, но нужно учитывать лимит запросов etherscan и добавить sleep после каждого блока), которая циклично будет делать запрос на АПИ etherscan и получать текущий блок и транзакции с него (методы которыми можно получить описаны ниже). После получения транзакций они должны сохраняться в MongoDB, откуда их уже будет получать пользователь через АПИ, а также при получении нового блока должно обновляться количество подтверждений на транзакциях которые были захвачены до этого.

3. При запуске приложения должны инициализироваться последние 1000 блоков с блокчейн (если нет транзакций в БД).
4. Данные по транзакциям можно получать из <https://docs.etherscan.io/api-endpoints/geth-parity-proxy> (будет достаточно методов `eth_getBlockByNumber` и `eth_getTransactionByHash`)

Требования к выполнению

Для работы со стилями использовать препроцессор SCSS

Страницы должны быть адаптивными

Для работы с базой данных использовать

<https://www.npmjs.com/package/mongoose>

Код разместить на github

Рабочий проект развернуть на heroku (или в любой другой песочнице).

Макеты:

<https://www.figma.com/file/LpWuCx7YkPctSdSVwAKfll/Untitled?node-id=8201%3A3>

для просмотра стилей и экспорта изображений нужно зарегистрироваться в figma