

Тестовое задание Front-end

Общие положения

Введение

Данное задание предназначено для выявления способностей и навыков Кандидата с точки зрения требований к работнику Лаборатории Распределенных Реестров на позиции Front-end разработчика.

Задание разработано исходя из регулярных практик и потребностей Лаборатории Распределенных Реестров в реализации программного обеспечения.

Общие требования к Кандидату

1. Внимательно изучить задание;
2. Выполнить тот объем задач, который Кандидат способен выполнить;
3. Главное – качество;
4. Обозначить срок выполнения задания;
5. При возникновении любых обстоятельств, препятствующих выполнению задания, уведомить ответственного сотрудника.

Порядок выполнения

1. Ознакомление с заданием;
2. Создание GitHub репозитория;
3. Вести разработку в ветке development;
4. Одна задача – один commit;
5. По готовности крупного раздела, merge в master;
6. Сверить работу по чек-листу;
7. Скопировать чек-лист, выделить зеленым готовый функционал, красным тот, что выполнить не удалось;
8. Направить задание на проверку.

Задание

Реализовать приложение с использованием технологий React, React-router, React-redux.

Запрещено использовать сторонние библиотеки, кроме material-ui.

Реализовать следующие элементы:

Навигационный бар

Описание:

Содержит ссылки на главную страницу, страницу с кошельками, страницу с настройками.

Поведение:

Отображается в приложении всегда.

Выделение акцентным цветом ссылки на текущую страницу.

На экранах менее 1280 px в ширину имеет вид burger-menu;

Главная страница

Описание:

Обозреватель постов.

Содержит постраничный навигатор, строку поиска, посты.

Поведение:

Отображает посты с fake-API <https://jsonplaceholder.typicode.com/posts>

Все посты выводятся по N штук на страницу, N задается настройками приложения.

Поиск срабатывает при отправке формы.

Поиск происходит по названию и по содержанию каждого из постов.

Страница с кошельками

Описание:

Страница показывает подключенные к приложению кошельки (их отсутствие) из сетей Ethereum и Cardano.

Поведение:

В каждом из разделов при отсутствии кошелька показывается кнопка вызова расширений браузера для подключения кошельков (для сети Ethereum использовать кошелек Metamask; для сети Cardano использовать кошелек Nami).

Раздел Cardano может быть выключен через настройки приложения.

При любом подключенном кошельке показывается адрес выбранного пользователем аккаунта.

Любой подключенный аккаунт (адрес) можно скопировать нажатием на соответствующую кнопку.

Каждый подключенный аккаунт (адрес) можно отключить нажатием на соответствующую кнопку.

При отключении аккаунта, состояние раздела сбрасывается.

Страница с настройками

Описание:

Настройка темы приложения

Настройка шрифта приложения

Настройка размера шрифта приложения

Настройка количества постов загружаемых на одну страницу

Настройка включения/выключения сети Cardano для приложения

Кнопка сброса настроек

Поведение:

Селектор темы меняет цветовую схему приложения.

Селектор шрифта меняет семейство шрифтов приложения.

Селектор размера шрифта меняет размер отображаемого шрифта приложения.

Настройка количества постов меняет количество отображаемых постов на главной странице.

Настройка сети Cardano включает/выключает возможность подключить кошелек из сети Cardano (можно выключить раздел, сделав его неактивным, отобразив сообщение о возможности включить в настройках).

Кнопка сброса настроек возвращает исходные значения.

Общие элементы

Описание:

Индикация загрузки

Поведение:

Отображает анимацию загрузки при выполнении всех асинхронных операций приложения.

Блокирует возможность взаимодействия с интерфейсом.

Чек-лист

Общие

- ☐ Имена переменных, функций и параметров написаны в camelCase.
- ☐ Имена переменных – существительные.
- ☐ Имена переменных, функций и компонентов соответствуют содержимому.
- ☐ Имена коллекций NodeList – существительные во множественном числе.
- ☐ Имя функции начинается с глагола и отражает то, что она делает.
- ☐ Для именования не используются транслит и неуместные сокращения.
- ☐ Код форматирован одинаково, соблюдается иерархия отступов.
- ☐ Хуки не используются внутри условных блоков или циклов.
- ☐ Хуки вызываются в основной функции компонента.
- ☐ При использовании классовых компонентов эффекты описаны внутри методов жизненного цикла компонента.
- ☐ Имена пользовательских хуков начинаются с use.
- ☐ Начальное состояние стейт-переменных содержит корректный тип данных.
- ☐ Для элементов списка используется уникальный ключ key, а не индекс массива.
- ☐ Запросы к API описаны в отдельном файле.
- ☐ Для внутренних ссылок в приложении используются компоненты из библиотеки react-router.
- ☐ Не происходит утечки памяти при навешивании обработчиков. Все обработчики, добавленные через addEventListener, удаляются при размонтировании компонента.
- ☐ Пользователь получает сообщение в случае любой ошибки.
- ☐ Поля формы заблокированы во время отправки запросов и у пользователя нет возможности отправить новый запрос до завершения предыдущего.
- ☐ Фиксированные значения (константы) именуются заглавными буквами и вынесены в отдельный конфиг-файл.
- ☐ Нет беспорядка в коде.
- ☐ Нет лишнего кода: переменная объявлена, но не используется или есть какая-то лишняя логика.
- ☐ Отсутствуют серьезные баги.

Верстка

- ☐ Использован примитивный дизайн, без излишне ярких цветов.

- ☐ Классы названы в соответствии с ролью элемента на странице.
- ☐ Соблюдается изолирование блоков, нет каши из элементов.
- ☐ Страница вписывается в подавляющее количество размерностей экранов.
- ☐ Используются семантические конструкции.
- ☐ Корректно использованы отступы и поля.
- ☐ Нет неиспользованных конструкций.

Навигация

- ☐ Все ссылки и кнопки в проекте работают.
- ☐ Правильно работает состояние навигационного бара: выделяется ссылка текущей страницы.
- ☐ Любую страницу приложения можно посетить через набор ссылки в строке браузера.
- ☐ Корректно работает функционал истории, работают кнопки назад/вперед.
- ☐ При попытке перейти на несуществующую страницу происходит редирект на страницу «404».

Главная страница

- ☐ Загрузчик работает во время запроса.
- ☐ Поиск регистронезависимый.
- ☐ Результаты поиска разбиваются на страницы.
- ☐ При отсутствии результатов, показывается информационное сообщение.
- ☐ После успешного сабмита формы поиска появляется блок с результатами.
- ☐ Количество страниц адаптируется под количество записей.
- ☐ Последний запрос кэшируется.
- ☐ При перезагрузке страницы состояние формы сохраняется.

Страница с кошельками

- ☐ Раздел с Cardano скрыт или неактивен, если выключен через настройки.
- ☐ При подключении кошелька происходит загрузка.
- ☐ Адрес кошелька отображен верно и совпадает с адресом из расширения(В случае с Cardano нужно добавить многоточие в середине адреса для его визуального сокращения).
- ☐ Визуально можно однозначно определить из какого расширения взят адрес.
- ☐ Адрес копируется по кнопке.

- ☐ Аккаунт(адрес) отсоединяется от приложения по кнопке.
- ☐ При перезагрузке страницы сохраняется состояние.
- ☐ При подключении из неправильной сети из расширения, выводится сообщение для пользователя из расширения на смену сети.
- ☐ Обработывается отказ пользователя подключить адрес к приложению.
- ☐ Выдается ошибка при несовпадении подключенного адреса и фактического.

Страница с настройками

- ☐ Все настройки пользователя хранятся в хранилище, состояние сохраняется при перезагрузке страницы
- ☐ Приложение содержит несколько тем, шрифтов, размеров шрифтов. Данные вынесены отдельно и управляются через выпадающий список.
- ☐ Тема, шрифт, размер шрифта меняется мгновенно, без подтверждения
- ☐ Раздел с Cardano применяется мгновенно.
- ☐ Количество постов на странице применяется мгновенно.
- ☐ Кнопка сброса работает корректно(Можно сделать сброс всех данных приложения, включая кошельки).
- ☐ Приложение содержит настройки по умолчанию.