

Тестовое задание для стажера React Frontend-разработчика

Стек технологий:

- TypeScript
- Webpack + Npm
- React 16
- Ant design 5 и styled-components
- React router 6
- TanStack Query 4 + axios
- dayjs
- Git (github.com)
- Архитектура FSD (Feature-Sliced Design)

Цель:

Реализовать, используя стек технологий, проект базы пользователей, содержащий основные компоненты:

- Страница авторизации
- Страница 404
- Страница пользователей
- Модальное окно редактирования пользователя
- Модальное окно создания пользователя

Требования:

1. Использовать api запросы, созданные с помощью бесплатного ресурса: mockapi.io
2. Проект должен разрабатываться в публичном github репозитории, для будущей проверки.
3. Для разработки использовать только список стека технологий

Этапы разработки:

Страница авторизации

- Использовать React Router 6 с route path – 'login'
- Пользователь с логином admin и паролем admin
- Для имитации авторизации создать промис функцию с таймаутом в 2000мс. При успешной авторизации возвращает (return resolve) токен (любую строку), при ошибках возвращает(return reject) объект new Error с текстом ошибки
- Для запроса использовать TanStack Query и axios
- Отображать ошибку (блоком или notification) с текстом ошибки, если пользователь ввел неправильные данные
- Блокировать кнопку авторизации, если идет загрузка
- При успешном логине записывать токен в localStorage
- При обновлении любой страницы проверять пользователя на авторизацию
- Если авторизованный пользователь перейдет на эту страницу - его перекинет на страницу пользователей

Страница 404

- Использовать React Router 6
- Использовать Result компонент antd (<https://ant.design/components/result#result-demo-404>)

Страница списка пользователей

- Использовать React Router 6 с route path – 'users'
- Для получения пользователей использовать api из mockapi.io и библиотеки TanStack Query и axios
- Кнопка «Выход» разлогинивает пользователя
- Кнопка создать пользователя показывает модальное окно создания пользователя
- Клик на аватар или имя показывает модальное окно редактирования пользователя
- Форматирование даты (Зарегистрирован) должно быть с помощью dayjs в виде DD.MM.YYYY

Модальное окно создания/редактирования пользователя

- Использовать компонент Modal из antd (<https://ant.design/components/modal>)
- Использовать api из mockapi.io
- Редактирование в отличие от создания содержит поле id, которое выключено для редактирования (disabled), но нужно для выполнения запроса редактирования
- Валидация на пустые поля
- Валидация на корректность ссылки
- Отключение кнопок сохранения/создания и «отмены» и возможности закрыть модальное окно при обращении к api
- Для сохранения и редактирования пользователя использовать TanStack Query и axios
- После сохранения или редактирования - обновить список пользователей

Порядок работы:

Подготовка mockapi.io:

- Регистрируемся с помощью гитхаба или гугла
- Создаем проект с любым именем
- Создаем ресурс (кнопка «New resource») с именем users (Названия полей и их тип оставить по умолчанию)

Resource name

Enter meaningful resource name, it will be used to generate API endpoints.

users

Schema

Define Resource schema, it will be used to generate mock data.

id

Object ID

createdAt

Faker.js

date.recent

name

Faker.js

name.fullName

avatar

Faker.js

image.avatar

- Скроля вниз модального окна, знакомимся с методами api, которые нужны будут для работы

On GET /users

\$mockData

On GET /users/:id

\$mockData

On POST /users

\$mockData

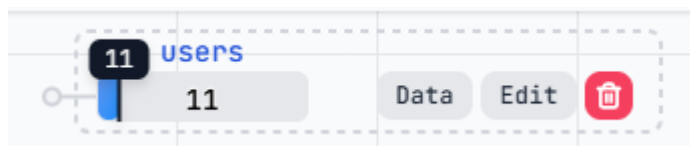
On PUT /users/:id

\$mockData

On DELETE /users/:id

\$mockData

- Добавляем любое количество пользователей кликом на слайдер



Репозиторий:

- Создать репозиторий на <https://github.com/> со своим стартовым шаблоном сборки либо сделать fork любого репозитория, который можно подстроить под стек технического задания
- Установить зависимости сборки
- Создать index.tsx файл в каталоге src, который будет являться точкой входа в приложение
- Создать каталоги для компонентов приложения. Архитектура – FSD.
- Для запуска разработки создать в package.json и использовать скрипт npm run start
- Разрабатывать приложение поэтапно: каждый этап 1 ветка с последующим мерджем в мастер ветку
- При разработке использовать изображения (мокапы) страниц и модальных окон.
- Итоговый результат в виде публичного репозитория github с приложением прислать ссылкой