

Тестовое задание на позицию frontend-разработчика в Volt.

Необходимо выполнить задание используя заготовку, доступную по ссылке:
<https://drive.google.com/file/d/0B9gbscdfWqaqOFJtWUs3YjZxOWc/view?usp=sharing>

Ссылку на репозиторий с выполненным заданием прислать на alex.ponomarev@wearevolt.com

Инициализация

Установка

```
npm install
```

Запуск бекенд-сервера

```
npm start
```

Доступ к приложению в браузере

```
http://localhost:8000
```

Общие требования

Задание заключается в реализации SPA-приложение на React.

Должны быть выполнены следующие условия:

- Приложение должно быть упаковано в JS-бандл с помощью webpack
- Приложение должно иметь навигацию с использованием History API
- В зависимости от выбранного интерфейса должен быть установлен соответствующий document title
- Интерфейсы должны быть сверстаны с применением react-bootstrap
- Выбор реализации data flow на ваше усмотрение

Основное задание. Интерфейсы покупателей и товаров.

Требуется реализовать следующие интерфейсы:

- Интерфейс покупателей (mockups/customers.png)

- Интерфейс товаров (mockups/products.png)

Каждый интерфейс должен быть реализован на отдельном маршруте роутера.

Интерфейс покупателей

- Реализовать список покупателей с добавлением/редактированием/удалением элементов (см. раздел Схема REST API -> Customers).
- Создание покупателя должно быть реализовано в модальном окне.
- Редактирование покупателя должно быть реализовано в модальном окне.
- Подтверждение удаления покупателя должно быть реализовано в модальном окне.

Интерфейс товаров

- Реализовать список товаров с добавлением/редактированием/удалением элементов (см. раздел Схема REST API -> Products).
- Создание товара должно быть реализовано в модальном окне.
- Редактирование товара должно быть реализовано в модальном окне.
- Подтверждение удаления товара должно быть реализовано в модальном окне.

Дополнительное задание.

Интерфейсы списка счетов и создания/редактирования счёта

Требуется дополнительно реализовать следующие интерфейсы:

- Интерфейс счетов (mockups/invoice-list.png)
- Интерфейс создания/редактирования счёта (mockups/invoice-edit.png)

Каждый интерфейс должен быть реализован на отдельном маршруте роутера.

Интерфейс счетов

- Реализовать список счетов с добавлением/редактированием/удалением элементов (см. раздел Схема REST API -> Invoices).
- Подтверждение удаления счёта должно быть реализовано в модальном окне.

Интерфейс создания нового/редактирования существующего счёта

Реализовать интерфейс создания/редактирования счета со следующим функционалом:

- установление размера скидки
- выбор покупателя
- добавление/редактирование количества/удаление товара
- подсчёт и сохранение суммы ИТОГО (total)

Структуры данных перечислены в разделах:

- Схема REST API -> Invoices
- Схема REST API -> InvoiceItems

Схема REST API

После запуска проекта (`npm start`) будет автоматически поднят-бекенд сервер реализующий следующие эндпоинты:

Customers

структура

- id (integer)
- name (string)
- address (string)
- phone (string)

эндпоинты

GET POST	/api/customers
GET PUT DELETE	/api/customers/{id}

Products

структура

- id (integer)
- name (string)

- price (decimal)

ЭНДПОИНТЫ

GET POST	/api/products
GET PUT DELETE	/api/products/{id}

Invoices

структура

- id (integer)
- customer_id (integer)
- discount (decimal)
- total (decimal)

ЭНДПОИНТЫ

GET POST	/api/invoices
GET PUT DELETE	/api/invoices/{id}

InvoiceItems

структура

- id (integer)
- invoice_id (integer)
- product_id (integer)
- quantity (decimal)

ЭНДПОИНТЫ

GET POST	/api/invoices/{id}/items
GET PUT DELETE	/api/invoices/{invoice_id}/items/{id}