

Модальное окно

Есть компонент модального окна. Необходимо:

- Отрендерить модальное окно в **App**
- Добавить в **App** состояние в котором храним флаг **isModalOpen**
- Добавить в **App** методы для открытия и закрытия модального окна
- Добавить в **App** кнопку для открытия модального окна

Модальное окно должно:

- Открываться по клику на кнопке в **App**
- Закрываться при клике по кнопке **Close** в модальном окне
- Закрываться при клике в **Backdrop** - тень модального окна перекрывающая фон при открытом окне
- Закрываться при нажатии **ESC** в окне

Используйте **refs** для получения ссылок на **Backdrop** и методы жизненного цикла для добавления слушателей на **window**.

```
// Modal.js
import React, { Component } from 'react';

const styles = {
  backdrop: {
    position: 'fixed',
    top: 0,
    left: 0,
    width: '100vw',
    height: '100vh',
    backgroundColor: 'rgba(0, 0, 0, 0.2)',
    display: 'flex',
    justifyContent: 'center',
    alignItems: 'center',
  },
  modal: {
    maxWidth: 600,
    minHeight: 300,
    backgroundColor: '#fff',
    padding: 16,
  },
};

export default class Modal extends Component {
  render() {
    const { onClose } = this.props;

    return (
      <div style={styles.backdrop}>
```

```
        <div style={styles.modal}>
          <p>
            Lorem ipsum dolor, sit amet consectetur adipisicing elit.
Nihil
            ipsum obcaecati maiores ipsam harum distinctio quia, soluta
            voluptatibus iste deserunt consectetur totam quas quidem,
aliquid
            voluptatem nisi, nobis expedita quis?
          </p>
          <button type="button" onClick={onClose}>
            Close
          </button>
        </div>
      </div>
    );
  }
}
```

CRUD истории заказов

Используя `json-server` поднимите локальный бекенд. Ниже `json` с начальными данными для `db.json`.

```
[
  {
    "id": 0,
    "date": "6/22/2018",
    "price": "290.00",
    "address": "108 Quinn Plains",
    "rating": 10
  },
  {
    "id": 1,
    "date": "5/15/2018",
    "price": "159.00",
    "address": "262 Jovan Roads",
    "rating": 4
  },
  {
    "id": 2,
    "date": "10/1/2018",
    "price": "710.00",
    "address": "331 Mills Mountains",
    "rating": 9
  },
  {
    "id": 3,
    "date": "5/12/2018",
    "price": "567.00",
  }
```

```
    "address": "47330 Boyle Road",
    "rating": 8
  },
  {
    "id": 4,
    "date": "9/22/2018",
    "price": "703.00",
    "address": "0700 Furman Trail",
    "rating": 6
  },
  {
    "id": 5,
    "date": "6/14/2018",
    "price": "887.00",
    "address": "24880 Margaret Harbors",
    "rating": 2
  }
]
```

- Используйте таблицу истории из домашней работы #1
- При рендере App, необходимо сделать запрос на бекенд и забрать список всех заказов, отрендерить его в таблице
- Добавьте в каждую строку таблицы кнопку "Удалить", при клике на которую, происходит HTTP-запрос на бекенд для удаления записи по id. После ответа удаляем на фронте.
- Добавьте в каждую строку таблицы кнопку "Детальнее", при клике на которую, идет HTTP-запрос на бекенд для получения записи по id. Результат ответа от бекенда отобразить в модальном окне в любом виде, можно просто JSON.parse, не важно!
- Когда идет запрос на одну запись по id, добавьте индикатор загрузки, который отображается пока идет запрос, после чего отображается модальное окно.
- Модальное окно можно закрыть и продолжить работать с таблицей.
- Добавьте простую форму для создания записи в истории. Дата используется на момент сабмита. Адрес доставки, цена и рейтинг вводятся в поля формы.
- При сабмите формы идет POST запрос на бекенд с составленным объектом. Ждем ответа от бекенда и если все ок, добавляем в список истории на фронте.