

**Московский государственный технический университет  
им. Н. Э. Баумана**

Факультет «Информатика и системы управления»

Отчёт по лабораторной работе №8  
по курсу «Разработка интернет-приложений»

Разработка пользовательского интерфейса с использованием библиотеки  
React.

Выполнил:

студент группы ИУ5-51Б  
Гришин С. В.

Проверил:

преподаватель каф. ИУ5  
Гапанюк Ю. Е.

Подпись и дата:  
26.12.2021

Подпись и дата:

Москва, 2021 г.

**Цель лабораторной работы:** изучение возможностей создания пользовательского интерфейса в веб-приложениях с использованием библиотеки React.

### Описание задания

1. Создать стартовый React-проект. Удалить неиспользуемый код. Организовать директории для страниц, компонентов, утилит и работы с сетью.
2. Организовать роутинг в веб-приложении.
3. Разработать базовые страницы, на которых будут отображаться сущности из выбранной вами предметной области:
  - i. Стартовая страница.
  - ii. Страница просмотра списка объектов.
  - iii. Страница просмотра конкретного объекта.
4. Вынести переиспользуемые компоненты в отдельные файлы:
  - i. Для навигации по приложению можно добавить header.
  - ii. Для отображения дополнительной информации (данные о студенте и предметной области) можно использовать footer.
  - iii. Источники ввода-вывода (поля ввода (inputs)/формы/текстовые блоки).
  - iv. Переиспользуемые таблицы/гриды.
5. Добавить асинхронные запросы в разработанный API, чтобы страница получала данные с сервера.
6. Если в Вашем проекте реализована сложная логика работы с состоянием приложения, то рекомендуется добавить пользовательские хуки.
7. Страницы приложения должны хорошо отображаться как на больших, так и на маленьких экранах.

### Ход выполнения лабораторной работы

App.js:

```
import { BrowserRouter, Route, Link, Switch } from "react-router-dom";
import StartPage from "../page";
import ITunesPage from "../modules/net";
import Page2 from "../modules/Itunes.js";
import net from "../modules/net";

function App() {

  return (
    <BrowserRouter basename="/" >
      <div>
        <ul>
          <li>
            <Link to="/">Главная страница</Link>

```

```

        </li>
        <li>
            <Link to="/country">Страница со странами</Link>

        </li>
    </ul>
    <hr />
    <Switch>
        <Route exact path="/">
            <h1>Это наша стартовая страница</h1>
        </Route>
        <Route path="/country" component={Page2}/>
        <Route path="/detail" component={net} />
    </Switch>
    </div>
</BrowserRouter>

);
}

export default App;

```

## Itunes.js:

```

import React from "react";
import {Button, Card, Col, Row} from "react-bootstrap";
import { BrowserRouter } from "react-router-dom";
import './Page2.css';
class Page2 extends React.Component {
    numb;
    Open(numb) {
        localStorage.setItem("key", 8)
        window.open("/detail", "_blank")
    }

    // Constructor
    constructor(props) {
        super(props);

        this.state = {
            items: [],
            DataisLoaded: false
        };
    }

    // ComponentDidMount is used to
    // execute the code
    componentDidMount() {
        fetch(
            "http://127.0.0.1:8000/countries/")
            .then((res) => res.json())
            .then((json) => {
                this.setState({
                    items: json,
                    DataisLoaded: true
                });
            })
    }

    render() {
        const {DataisLoaded, items} = this.state;
        if (!DataisLoaded) return <div>
            <h1> Pleses wait some time.... </h1></div>;

        return (
            <BrowserRouter basename="/new">

```

```

        <div className="App">
          <h1> Страны </h1> {
            <Row xs={4} md={4} className="g-4">
              {items.map((item, index) => {
                return <Col>
                  <Card key={index} className="card" id="xd">
                    <div className="lol">
                      <Card.Img className="cardImage"
variant="top" src={item.src} height={200}
                                width={300}/>
                      <Card.Body>
                        <div className="textStyle">
<Card.Title>{item.title}</Card.Title>
                                </div>
                                <div className="textStyle">
                                  <Card.Text>
                                    {item.text}
                                  </Card.Text>
                                </div>
                                <Button className="cardButton"
onClick={() => this.handleRegionClick(item.pk)}>Открыть</Button>
                                  </Card.Body>
                                </div>
                              </Card>
                            </Col>
                          )}
            </Row>
          }
        </div>
      </BrowserRouter>
    );
  }

  handleRegionClick(id) {
    localStorage.setItem("key", id)
    window.open("/detail", "_blank")
  }
}

export default Page2;

```

Page2.css:

```

.Page2 {
  text-align: center;
  color: Green;
}

.Page2-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}

.App {

```

```

        display: inline-block;
        text-align: center;
    }
    .lol{
        display: inline-block;
        text-align: center;
        border: 1px solid black;
        margin: 5px;
    }
    #xd{
        display: inline-block;
    }
    .Page2-link {
        color: #61dafb;
    }

    @keyframes Page2-logo-spin {
        from {
            transform: rotate(0deg);
        }
        to {
            transform: rotate(360deg);
        }
    }
}

```

## App.css:

```

.App {
    text-align: center;
}

.App-logo {
    height: 40vmin;
    pointer-events: none;
}

@media (prefers-reduced-motion: no-preference) {
    .App-logo {
        animation: App-logo-spin infinite 20s linear;
    }
}

.App-header {
    background-color: #282c34;
    min-height: 100vh;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    font-size: calc(10px + 2vmin);
    color: white;
}

.App-link {
    color: #61dafb;
}

@keyframes App-logo-spin {
    from {
        transform: rotate(0deg);
    }
    to {
        transform: rotate(360deg);
    }
}

```

```
}  
}
```

Net.js:

```
import React from "react";  
import {Card, Col, Row} from "react-bootstrap";  
import './Page2.css';  
class Page2 extends React.Component {  
  
  // Constructor  
  numb;  
  constructor(props) {  
    super(props);  
  
    this.state = {  
      items: [],  
      DataisLoaded: false  
    };  
  }  
  
  // ComponentDidMount is used to  
  // execute the code  
  componentDidMount() {  
    fetch(  
      "http://127.0.0.1:8000/countries/"  
    ).then((res) => res.json())  
    .then((json) => {  
      this.setState({  
        items: json,  
        DataisLoaded: true  
      });  
    })  
  }  
  
  render() {  
    this.numb = localStorage.getItem("key");  
    console.log("Внимание", this.numb);  
    const { DataisLoaded, items } = this.state;  
    if (!DataisLoaded) return <div>  
      <h1> Pleses wait some time.... </h1> </div> ;  
  
    return (  
      <div className = "App">  
        <h1>Детальная информация о стране</h1> {  
        <Row xs={4} md={4} className="g-4">  
          {items.map((item, index)=>{  
            if(item.pk==this.numb)  
              return<Col >  
                <Card key={index} className="card">  
                  <Card.Img className="cardImage"  
variant="top" src={item.src} height={500} width={700} />  
                  <Card.Body>  
                    <div className="textStyle">  
  
<Card.Title>{item.title}</Card.Title>  
  
                    </div>  
                    <div className="textStyle">  
                      <Card.Text>  
                        {item.text}  
                      </Card.Text>  
                    </div>  
                  </Card.Body>  
                </Card>  
              }  
            }  
          )}  
        </Row>  
      </div>  
    )  
  }  
}
```

```

        </Col>
      )} }
    </Row>
  }
</div>
);
}
}

export default Page2;

```

page.js:

```

import React, {useEffect, useState} from 'react';

const data = [
  'Берик Дондаррион',
  'Леди Мелиссандра',
  'Полливер',
  'Уолдер Фрей',
  'Тайвин Ланнистер',
  'Сир Мерин Трэнт',
  'Король Джоффри',
  'Сир Илин Пейн',
  'Гора',
  'Пес',
  'Серсея Ланнистер',
]

function StartPage() {

  // В функциональных компонентах для работы с состоянием можно
  использовать хук useState()
  // Он возвращает кортеж из двух элементов:
  // 1 элемент - объект состояния
  // 2 элемент - метод который позволит нам обновить состояние
  const [randomName, setRandomName] = useState();

  // Кстати, это хороший пример деструктуризации массива в JavaScript
  const [names, setNames] = useState(data);

  const [showNames, setShowNames] = useState(false);

  // В данном хендлере мы изменяем состояние на какое-то конкретное
  const handleShowNames = () =>{
    setShowNames(true)
  }

  // В данном хендлере мы изменяем состояние на какое-то конкретное
  const handleHideNames = () =>{
    setShowNames(false)
  }

  // В данном хендлере мы изменяем состояние в зависимости от его прошлого
  значения
  const handleSwitch = () =>{
    // метод из useState может принимать как определенное значение, так и
    метод,
    // принимающий прошлое значение и возвращающий новое
    setShowNames(state => !state)
  }

  useEffect(()=>{

```

```

        console.log('Этот код выполняется только на первом рендере
компонента')
        // В данном примере можно наблюдать Spread syntax (Трехточие перед
массивом)
        setNames(names=>[...names, 'Бедный студент'])

        return () => {
            console.log('Этот код выполняется, когда компонент будет
размонтирован')
        }
    }, [])

    useEffect(() => {
        console.log('Этот код выполняется каждый раз, когда изменится
состояние showNames ')
        setRandomName(names[Math.floor(Math.random()*names.length)])
    }, [showNames])

    return (
        <div>
            <h3>Случайное имя из списка: {randomName}</h3>
            { /*Кнопка для того, чтобы показать имена*/ }
            <button onClick={handleShowNames}>Хочу увидеть список
имен</button>
            { /*Кнопка для того, чтобы скрыть имена*/ }
            <button onClick={handleHideNames}>Хочу скрыть список
имен</button>

            { /*Универсальная кнопка*/ }
            <button onClick={handleSwitch}>{showNames ? 'Хочу скрыть список
имен' : 'Хочу увидеть список имен' }</button>

            { /*React отрисует список только если showNames будет равен true,
boolean значения игнорируются при отрисовке*/ }
            {showNames && <ul>
                { /*Рекомендую почитать про прекрасные встроенные методы
массивов в JavaScript */ }
                {names.map((name, index)=>{
                    return (
                        <li key={index}>
                            <span>{name}</span>
                        </li>
                    )
                })}
            </ul>
            }
        </div>
    );
}

export default StartPage;

```

## Скриншоты результата



- [Главная страница](#)
- [Страны по странам](#)

Это наша стартовая страница

Назад



Россия

145 975 300 чел.