



**Министерство науки и высшего образования
Российской Федерации Федеральное
государственное бюджетное образовательное
учреждение высшего образования
«Московский государственный
технический университет имени Н.Э.
Баумана
(национальный
исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

ФАКУЛЬТЕТ Информатика и системы управления
КАФЕДРА Системы обработки информации и управления

**Лабораторная работа №8
По курсу «Разработка интернет приложений»**

Подготовил:
Студент группы ИУ5-55Б.
Турчин Д.С.
18.12.2020

Проверил:
Преподаватель кафедры ИУ5
Гапанюк Ю.Е.

Москва, 2021 г.

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

Задание:

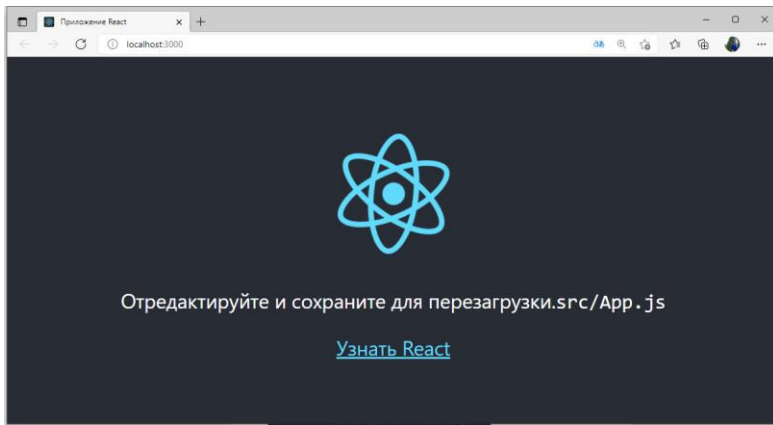
Разработать React-приложение. Для создания приложения необходимо решить следующие задачи:

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 работы

1) Создание React-приложения

```
npx create-react-app lab8  
cd lab8  
npm start
```



Создать оптимизированную сборку приложения:
`npm run build`

Запустить локальный сервер для разработки:
`npm run start`

Почистим ненужные файлы и организуем структуру папок. Получим следующую структуру приложения, аналогичную 7 ЛР:

2) Роутинг в react приложении

Установим библиотеки `react-router-dom` и `react-router`
`npm install react-router react-router-dom`

Сделаем так, что `App.js` будет отвечать за роутинг в нашем приложении. Он же отвечает за загрузку данных с АПИ-сервера при запуске приложения.

```

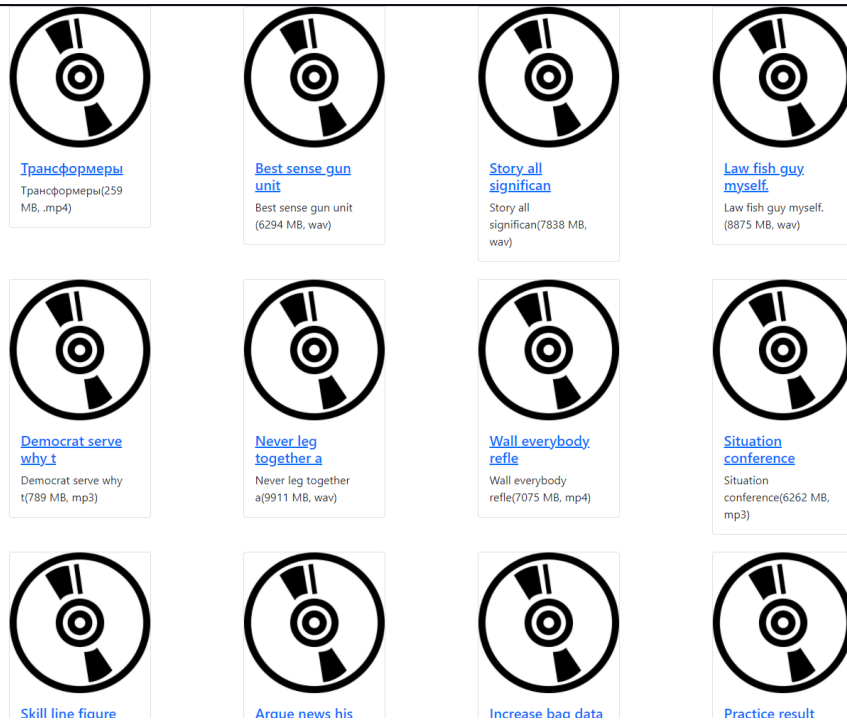
JS App.js 1 ●
lab-8 > app > src > JS App.js > App
17   setLoading(true);
18   ajax.get({url: urls.disks()}).then(({response}) => {
19     setDisks(response);
20     setLoading(false);
21   });
22 };
23
24   useEffect(() => {
25     // код выполнится на mount`е компонента
26     upload();
27     return () => {
28       // код выполнится на unmount`е компонента
29     }
30   },
31   [] // Это список зависимостей хука, он будет вызван каждый раз, когда зависимости будут меняться
32 );
33
34   return (
35     <BrowserRouter basename="/">
36     <div>
37       <Routes>
38         {/* <Route path="/" element={<StartPage/>}/> */}
39         <Route path="/" element={<MainPage loading={loading} disks={disks}/>}/>
40         {/* <Route path="/new" element={<NewPage/>}/> */}
41         <Route path="/:diskId" element={<DiskPage disks={disks}/>}/>
42       </Routes>
43     </div>
44   </BrowserRouter>
45 );
46 }
47
48 export default App;

```

Роутер позволяет нам перемещаться между разными страницами приложения без перезагрузки.

3) Базовые страницы приложения

- Создадим директорию со страницами нашего приложения src/pages
- Создадим стартовую страницу с дисками



```
JS Appjs 1 ● mainPage.jsx X
lab-8 > app > src > pages > main > mainPage.jsx > MainPage
1 import {Col, Row, Spinner} from "react-bootstrap";
2 import DiskCard from '../components/diskCard/diskCard.jsx';
3 // import {useState} from 'react';
4
5 function MainPage({loading, disks}) {
6   return (
7     <div className={`container ${loading && 'containerLoading'}`}>
8       {loading && <div className="loadingBg"><Spinner animation="border"/></div>}
9
10      <Row xs={4} md={4} className="g-4">
11        {disks.map((item, index) =>
12          <Col key={index}>
13            <DiskCard {...item} />
14          </Col>
15        )}
16      </Row>
17    </div>
18  );
19 }
20
21 export default MainPage;
22
```

- Создадим страницу просмотра конкретного диска

lab-8 > app > src > pages > disk > diskPage.jsx > DiskPage

```
1 import diskDefaultImg from '../static/img/disk-icon.png';
2 import {Card, Button} from "react-bootstrap";
3 import {useParams} from "react-router-dom";
4
5 const DiskPage = ({disks}) => {
6   const {diskId} = useParams();
7   console.log({diskId}, typeof diskId);
8   const disk = disks.find((el) => el.id + '' === diskId);
9   if (!disk) {
10     return null;
11   }
12   return <div>
13     <Button href="/">Назад</Button>
14     <Card className="card" style={{width: '200px'}}>
15       <Card.Img className="card-img-top" src={diskDefaultImg} alt={disk.title}/>
16       <Card.Body>
17         <div className="textStyle">
18           <Card.Title>{disk.title}</Card.Title>
19         </div>
20         <div className="textStyle">
21           <Card.Text>
22             {disk.description}
23           </Card.Text>
24         </div>
25       </Card.Body>
26     </Card>;
27   </div>
28 }
29 export default DiskPage;
```



Law fish guy myself.

Совсем другая верстка! Law fish guy
myself.(8875 MB, wav)

[Назад](#)