

Лабораторная работа: клиентская маршрутизация

Упражнение 1: Создание маршрутизации

1.1. В файле **src\App.css** создайте оформление для навигационного меню

```
ul {  
  list-style-type: none;  
}  
.App {  
  width: 1000px;  
  margin: auto;  
}  
.App-nav ul {  
  display: flex;  
  width: 100%;  
  justify-content: flex-start;  
}  
.App-nav a {  
  display: block;  
  padding: 5px 10px;  
  background: rgb(244, 244, 244);  
  text-decoration: none;  
  text-transform: uppercase;  
}
```

1.2. В **src\App.js** сразу после `<header className="App-header">` добавьте пункты меню

```
<nav className="App-nav">  
  <ul>  
    <li>  
      <a href="#" onClick={clickTopMenuHandler} data-route={"/"}>  
        Главная  
      </a>  
    </li>  
    <li>  
      <a href="#" onClick={clickTopMenuHandler} data-route={"/basket"}>  
        Корзина  
      </a>  
    </li>  
    <li>  
      <a href="#" onClick={clickTopMenuHandler} data-route={"/about"}>  
        О нас  
      </a>  
    </li>  
    <li>  
      <a href="#" onClick={clickTopMenuHandler} data-route={"/react-router"}>  
        О рутинге  
      </a>  
    </li>  
  </ul>  
</nav>
```

```

    </a>
  </li>
</ul>
</nav>

```

1.3. Опишите обработчик нажатия на пункт меню `clickTopMenuHandler`:

```

const [currentPath, setCurrentPath] = useState("/");

const clickTopMenuHandler = (ev) => {
  ev.preventDefault();

  let path = ev.target.getAttribute("data-route");
  console.log(`нажали для перехода ${path}`);

  const state = {};
  const title = "";

  window.history.pushState(state, title, path);

  setCurrentPath(path);
};

```

1.4. Создайте переменные `title` (содержание заголовка) и `content` (основное содержимое страницы). Заполните их значениями в зависимости от значения текущего пути `currentPath`

```

let title = null;
let content = null;

switch (currentPath) {
  case "/":
    title = "Каталог товаров";
    content = <div>Тут может быть перечень товаров</div>;
    break;
  case "/basket":
    title = "Корзина";
    content = basketPlace;
    break;
  case "/about":
    title = "О нас";
    content = <div>Тут информация о компании</div>;
    break;
  case "/react-router":
    title = "React-router";
    content = (
      <div>
        Часто используются не самописные, а сторонние системы рутинга, например{"
      <a href="https://reactrouter.com/" target="_blank">

```

```
        React Router
      </a>
    </div>
  );
  break;
}
```

Выводы

- посмотрели пример организации маршрутизации
- меню и ссылки можно оформить отдельными компонентами
- можно создать отдельные компоненты страницы
- наш вариант маршрутизации не учитывает передачу GET-параметров
- наш вариант маршрутизации не динамические фрагменты маршрута
- для реализации маршрутизации часто используется [React Router](#)