

Урок 5.6. React Router

Установка

```
npm install react-router-dom@6
```

Инициализация

Для того, чтобы роутер заработал, необходимо добавить компонент `<BrowserRouter>` в `index.js` (который находится в `src`)

```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css'; import App from './App';
// Импорт BrowserRouter
import {BrowserRouter} from 'react-router-dom';

ReactDOM.render
( <React.StrictMode>
  { /* Оборачиваем App в BrowserRouter */ }
  <BrowserRouter>
    <App />
  </BrowserRouter>
</React.StrictMode>, document.getElementById('root')
);
```

Создание списка роутов

```
import './App.css';
import Header from "../components/Header/Header";
import Footer from "../components/Footer/Footer";
import {Routes, Route, Outlet} from "react-router-dom";
import IndexPage from "../components/IndexPage/IndexPage";
import AboutPage from "../components/AboutPage/AboutPage";
import ContactsPage from "../components/ContactsPage/ContactsPage";
import PortfolioPage from "../components/PortfolioPage/PortfolioPage";
import UsersPage from "../components/UsersPage/UsersPage";
import UserPage from "../components/UserPage/UserPage";

function App() {
  return (
    <Routes>
```

```

        {/* По основному роуту всегда загружается Layout, который содержит всё необходимое для каждого из внутренних роутов (хедер, футер) */}
        <Route path={'/'} element={<Layout/>}>
            {/* index - это главный роут, он совпадает с роутом родительским */}
            {/* В данном случае IndexPage будет загружаться при корневом роуте */}
            <Route index element={<IndexPage/>}/>
            {/* Внутренние роуты: в строке браузера localhost:3000/about, localhost:3000/portfolio ... */}
            <Route path={'about'} element={<AboutPage/>}/>
            <Route path={'portfolio'} element={<PortfolioPage/>}/>
            <Route path={'contacts'} element={<ContactsPage/>}/>

            {/* Роут, который будет содержать в себе ещё дочерние роуты, */}
            {/* также можно задать через Layout (можно сделать его пустым, главное, */}
            {/* чтобы там был компонент <Outlet/> */}
            <Route path={'users'} element={<UsersLayout/>}>
                {/* Главный роут, загрузится по адресу localhost:3000/users (совпадает с родительским) */}
                <Route index element={<UsersPage/>}/>
                {/* Роут загрузится по адресу localhost:3000/users/any-string, где any-string - любая строка */}
                {/* any-string - это параметр роута, и он попадёт в userId */}
                <Route path={'/:userId'} element={<UserPage/>}/>
                {/* Роут загрузится по адресу localhost:3000/users/static (здесь имя задано жёстко) */}
                <Route path={'static'} element={<UserPage/>}/>
            </Route>
        </Route>
    </Routes>
);
}

function UsersLayout() {
    return (
        // Layout пользователя. Работает для страниц UsersPage и UserPage
        // Не содержит ничего, кроме Outlet (чтобы роутер знал, куда вставлять контент страниц)
        <Outlet/>
    )
}

function Layout() {
    return (
        // Основной Layout. Содержит основные элементы страницы, которые
        // загружаются на все роуты
        <div className="App">
            <Header/>
            <main className={'content'}>
                {/* Outlet показывает, куда нужно вставлять дочерние страницы */}
                <Outlet/>
            </main>
            <Footer/>
        </div>
    )
}

```

```
}  
  
export default App;
```

Получение параметров роута в странице

```
import {useParams} from "react-router-dom";  
  
function UserPage() {  
  // Получили параметр userId  
  // Имя совпадает с именем параметра в теге:  
  // <Route path=:userId" ... />  
  const { userId } = useParams();  
  return (  
    <h1>Страница конкретного пользователя {userId}</h1>  
  )  
}  
  
export default UserPage;
```