**React Router**

**Статические роуты**

Установка :

**npm install react-router-dom**

(yarn react-router-dom)

--------------------------------------------index.js

Импортировать компонент BrowserRouter из 'react-router-dom' и «обернуть» этим компонентом всё приложение :

import React from 'react';

import ReactDOM from 'react-dom';

import App from './App';

**import {BrowserRouter} from 'react-router-dom';**

const application = (

<**BrowserRouter**>

<App />

</**BrowserRouter**>

)

ReactDOM.render(

application,

document.getElementById('root')

);

--------------------------------------------------

В компонент, где находятся ссылки на страницы Home и Cars, в данном случае это App, импортируем компонент Route из 'react-router-dom' и с его помощью описываем роуты для наших ссылок.

В примере показаны два варианта описания роута : с использование служебного слова render и с использованием служебного слова component.

Варианты равнозначны.

Служебное слово exact контролирует точное совпадение пути, указанного в path.

--------------------------------------------App.js

import React, {Component} from 'react'

**import {Route} from 'react-router-dom'**

import Cars from './Cars/Cars'

class App extends Component {

render() {

return (

<div>

<nav className="nav">

<ul>

<li>

<a href="/">Home</a>

</li>

<li>

<a href="/cars">Cars</a>

</li>

</ul>

</nav>

**<Route path="/" exact render={() => <h1>Home Page</h1>} />**

**<Route path="/cars" component={Cars} />**

</div>

);

}

}

export default App;

--------------------------------------------------

Теперь роуты работают, но переход происходит с перезагрузкой страниц.

Чтобы устранить этот недостаток, добавим в импорт компонент NavLink. Заменим все ссылки (все теги <a>) специальними ссылками NavLink и href заменим на to. Используем служебное слово exact для контроля точного совпадения пути :

--------------------------------------------App.js

import React, {Component} from 'react'

import {Route, **NavLink**} from 'react-router-dom'

import Cars from './Cars/Cars'

class App extends Component {

render() {

return (

<div>

<nav className="nav">

<ul>

<li>

**<NavLink exact to="/">Home</NavLink>**

</li>

<li>

**<NavLink to="/cars">Cars</NavLink>**

</li>

</ul>

</nav>

<Route path="/" exact render={() => <h1>Home Page</h1>} />

<Route path="/cars" component={Cars} />

</div>

);

}

}

export default App

--------------------------------------------------

По умолчанию браузер помечает активную сылку классом “active”. Этот класс можно использовать для стилизации активной ссылки. Например в файле App.css

Есть возможность поменять имя класса активной ссылки :

<NavLink

to="/"

exact

**activeClassName={'new-active'}**

>Home</NavLink>

Есть возможность напрямую указать стиль активной ссылки :

<NavLink

to="/cars"

**activeStyle={{ color: 'blue' }}**

>Cars</NavLink>

Компонент NavLink позволяет конфигурировать url :

<NavLink **to={{**

**pathname: '/cars',**

**search: '?a=1&b=2',**

**hash: 'my-hash'**

**}}**>Cars</NavLink>

--------------------------------------------------