

<TeachMeSkills />

42. Навигация сайта в React

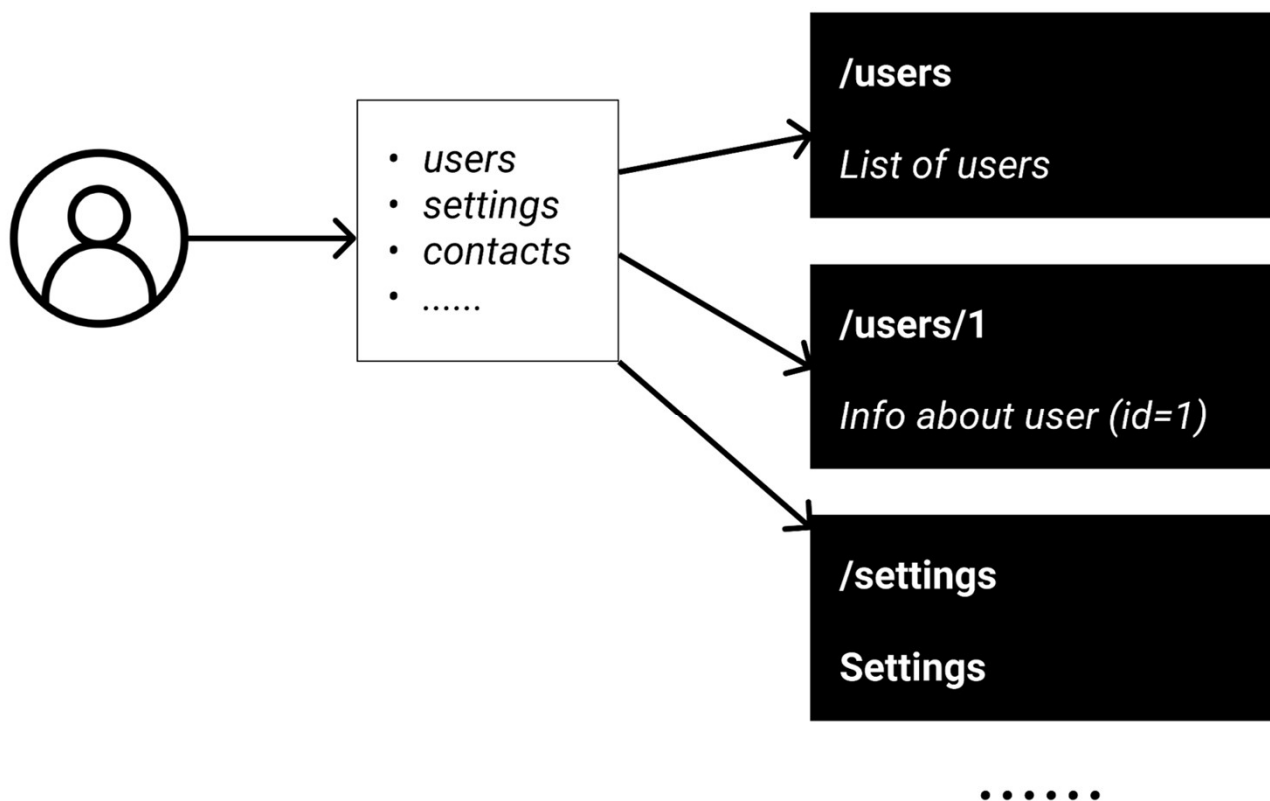
Цель:

Познакомиться со следующими понятиями и возможностями react:

- маршрутизация сайта
- что такое router

Маршрутизация:

Маршрутизация необходима, чтобы пользователь видел где он находится в данный момент времени в приложении, а также иметь возможность навигации по истории.



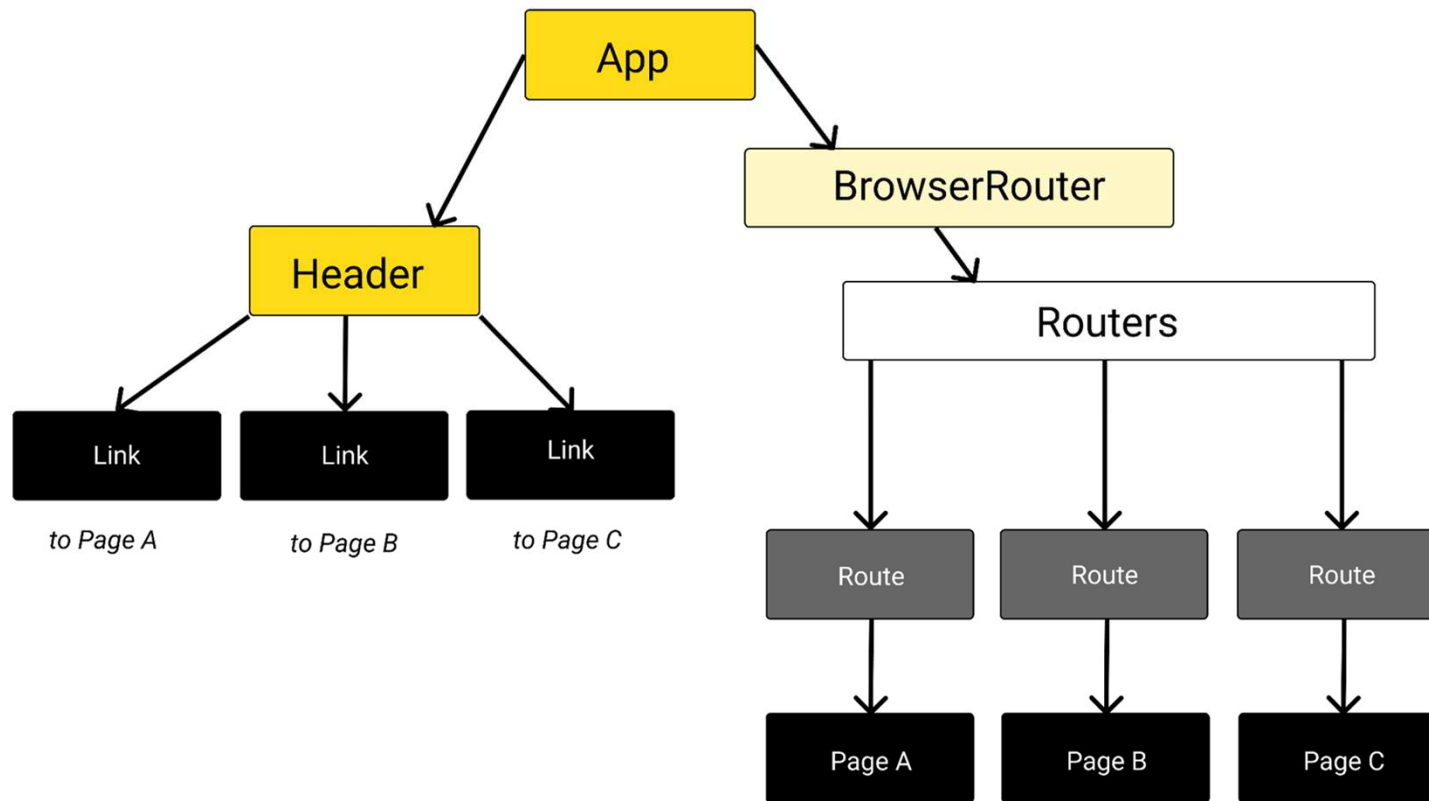
Маршрутизация:

Так как сам реакт не дает инструментов для маршрутизации, это можно решить за счет дополнительных библиотек.

react-router-dom



react-router-dom:



Роутер:

BrowserRouter определяет набор маршрутов и, когда к приложению, приходит запрос, то Router выполняет сопоставление запроса с маршрутами. И если какой-то из маршрутов совпадает с URL запроса, то этот маршрут выбирается для обработки запроса.

Routes - объект, содержащий в себе набор маршрутов. Он позволяет выбрать первый попавшийся правильный маршрут и использовать его для обработки.

Route определяет зависимость между URL и компонентом.

Конфигурация Routers:

```
1  import { render } from "react-dom";
2  import {
3    BrowserRouter,
4    Routes,
5    Route
6  } from "react-router-dom";
7  // import your route components too
8
9  render(
10    <BrowserRouter>
11      <Routes>
12        <Route path="/" element={<App />} />
13        <Route index element={<Home />} />
14        <Route path="teams" element={<Teams />} />
15          <Route path=":teamId" element={<Team />} />
16          <Route path="new" element={<NewTeamForm />} />
17          <Route index element={<LeagueStandings />} />
18        </Route>
19      </Route>
20    </Routes>
21  </BrowserRouter>,
22  document.getElementById("root")
23 );
```

Особенности Route:

- **element:** тот компонент, который отвечает за обработку запроса по этому маршруту
- **path:** шаблон адреса, с которым будет сопоставляться запрошенный адрес URL element
- **path="*":** Путь в виде звездочки - "*" в **path** указывает, что этот маршрут будет сопоставляться со всеми адресами URL, которые не соответствуют предыдущим маршрутам.
- **exact:** Атрибут exact используется в <Route>, чтобы сказать данный <Route> работает только если URL на браузере абсолютно подходит значению атрибута его path.

Вложенные роуты:

"/invoices"

"/invoices/sent"

"/invoices/:invoiceId"

```
1  function App() {  
2    return (  
3      <Routes>  
4        <Route path="invoices" element={<Invoices />}>  
5          <Route path=":invoiceId" element={<Invoice />} />  
6          <Route path="sent" element={<SentInvoices />} />  
7        </Route>  
8      </Routes>  
9    );  
10 }
```

Ссылки: Link & NavLink:

Link и NavLink
применяются для
создания ссылок.

```
1  import { Link } from "react-router-dom";
2
3  function Home() {
4    return (
5      <div>
6        <h1>Home</h1>
7        <nav>
8          <Link to="/">Home</Link> |{" "}
9          <Link to="about">About</Link>
10        </nav>
11      </div>
12    );
13  }
```

History (useHistory):

History - это объект, где хранится наша история навигации по сайту (стек маршрутов), а также методы для удобной работы с историей.

Хук **useHistory** позволяет нам получить доступ к объекту history.

```
import { useHistory } from "react-router-dom";

function HomeButton() {
  let history = useHistory();

  function handleClick() {
    history.push("/home");
  }

  return (
    <button type="button" onClick={handleClick}>
      Go home
    </button>
  );
}
```

Методы History:

- **goBack** позволяет перенаправить пользователя к предыдущему маршруту в стеке истории. Например, если пользователь перейдет со страницы Home на страницу Students, а затем нажмет кнопку для возврата назад (“Go Back”), он снова будет перенаправлен на страницу Home.
- **push** позволяет добавлять новые записи в стек истории и заставляет пользователя перейти на этот маршрут
- **location** - объект, содержащий в себе информацию о текущем месторасположении. Может включать в себя следующие свойства: `pathname`, `search`, `hash`, `state`