

42. Навигация сайта в React

Цель:

Познакомиться со следующими понятиями и возможностями react:

- маршрутизация сайта
- что такое router

План занятия:

- Что маршрутизация сайта и зачем она нужна
- Что такое router
 - Route
 - Ссылки: Link, NavLink
 - параметры маршрутов
 - Navigate
- History
- Private/public route

Конспект:

Маршрутизация:

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

Приложение должно уметь сопоставлять определённый URL с соответствующей ему страницей. То есть, если мы введем в адресную строку например `https://some-site.com/students`, то приложение должно направить нас на страницу списка студентов, но не на какую-либо другую.

Так как сам реакт не дает инструментов для маршрутизации, это можно решить за счет дополнительных библиотек. Например: `react-router`, `aviator` и др.

Больше популярных библиотек можно найти на официальном сайте: <https://ru.react.js.org/community/routing.html>

Библиотека `react-router-dom` (`BrowserRouter`, `Routes`, `Route`):

`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.

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 браузер абсолютно подходит значению атрибута `path`.

Ссылки: **Link & NavLink**

Link и **NavLink** применяются для создания ссылок. **NavLink** объект во многом похож на **Link** за тем исключением, что позволяет использовать состояние ссылки. Например, с помощью атрибутов `className` и `style` можно установить стиль активной ссылки.

Параметры маршрутов (**useParams**)

Для получения параметров маршрута применяется хук `useParams`. Полученное значение **params = useParams()** - это объект, где каждый отдельный параметр - это свойство этого объекта. Чтобы получить параметр `id`, необходимо использовать выражение `params.id`.

Редирект (**Navigate, useNavigate**)

Navigate используется для переадресации в зависимости от некоторого условия (условная переадресация).

useNavigate используется для программной переадресации.

History

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

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

Хук **useHistory** позволяет нам получить доступ к объекту history. И дает доступ к методам объекта history, как goBack или push.

Public/private route:

<https://medium.com/@thanhbinh.tran93/private-route-public-route-and-restricted-route-with-react-router-d50b27c15f5e>