

⚡ Current Skill Router Hooks

Router hooks

The react router library embrace the power of hooks, since the fifth version and introduce four different hooks to help our routing.

- `useNavigate()`
- `useLocation()`
- `useParams()`
- `useRouteMatch()`

We are going to explore each one of these hooks in the next slides

`useNavigate()` and `useLocation()`

- `useNavigate` is a hook that return a function allowing us to change the url whenever we want. It gives the same the result as the `Link` but it the `navigate` function can be used inside another function.

Let's see the example below, where the profile page will navigate to the home 5 seconds



```
import { useEffect, useState } from 'react';
import { Route, Routes, useNavigate } from 'react-router-dom';
import './App.css';

const Home = () => {
  const navigate = useNavigate();

  return (
    <div>
      <h1>
        Home page
      </h1>
      { /*
```



When we click this button the navigate function will change the url to

`/profile` and render the profile component

```
    */}

    <button onClick={() => navigate('/profile')}>Navigate to Profile </button>

  </div>
)
}

const Profile = () => {
  const navigate = useNavigate();
  const [timer, setTimer] = useState(0)
  useEffect(() => {
    setInterval(() => {
      setTimer(timer + 1)
    }, 1000);
  })
  /*
  After the 5 second the function navigate will change the url into `/`
  and the home page will be rendered
  */
  useEffect(() => {
    setTimeout(() => {
      navigate('/')
    }, 5000);
  }, [])
  return (
    <div>
      <h1>Profile Page</h1>
      <p>
        Count down: <span>{timer}</span>
      </p>
    </div>
  )
}

function App() {
```

```

return (
  <div className="App">
    <Routes>
      <Route path="/" element={<Home />}></Route>
      <Route index path="/profile" element={<Profile />}></Route>
    </Routes>
  </div>
);
}

```

```
export default App;
```

- **useLocation** : This hook returns the location object used by the react-router. This object represents the current URL and is immutable. Whenever the URL changes, the `useLocation()` hook returns a newly updated location object. Some of its use includes extracting the query parameters from the URL

In the example below, we create a new route **about** where the element to render is the About page

```

import { useEffect } from 'react'
import { useLocation } from 'react-router-dom'

const About = () => {
  const location = useLocation()
  useEffect(() => {
    console.log("🚀 ~ file: About.js ~ line 5 ~ About ~ history", location)
  }, [])
  return (
    <div>
      <h1>
        About Page
      </h1>
    </div>
  )
}

```

export default About

The output of the useEffect function is the image below:

```
🚀 ~ file: About.js ~ line 5 ~ About ~ history
▼ {pathname: '/about', search: '', hash: '', state: null, key: 'default'} ⓘ
  hash: ""
  key: "default"
  pathname: "/about"
  search: ""
  state: null
  ► [[Prototype]]: Object
```

useParams() and useMatchRoute()

- useParams(): This hook returns an object that consists of all the parameters in URL.

In the example below, we navigate to `/profile/admin` where admin is the id of user

```
import { useEffect } from 'react';
import { Route, Routes, useNavigate, useParams } from 'react-router-dom';
import './App.css';

const Home = () => {
  const navigate = useNavigate();

  return (
    <div>
      <h1>
        Home page
      </h1>
      <button onClick={() => navigate('/profile/John')}>Navigate to Profile </button>
    </div>
  )
}

const Profile = () => {
  const params = useParams();

  useEffect(() => {
    console.log("🚀 ~ file: App.js ~ line 23 ~ Profile ~ params", params)
```

```

    })

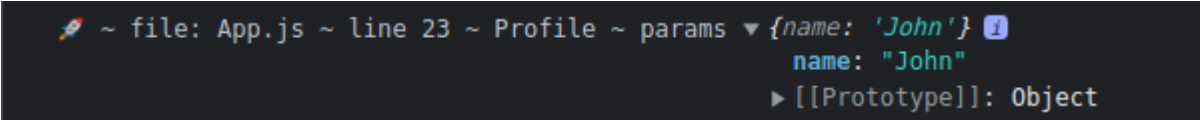
    return (
      <div>
        <h1>Profile Page</h1>
        <p>
          this page is for {params.name}
        </p>
      </div>
    )
  }
}

function App() {
  return (
    <div className="App">
      <Routes>
        <Route path="/" element={<Home />}></Route>
        {/* Here the parameter is the url is called `name`
          so the param that will be injected in useParams hook will be `name` */}
        <Route index path="/profile/:name" element={<Profile />}></Route>
      </Routes>
    </div>
  );
}

export default App;

```

The output of `console.log` will be the below:



```

~ file: App.js ~ line 23 ~ Profile ~ params {name: 'John'}
  name: "John"
  [[Prototype]]: Object

```

- `useMatchRoute`: Returns match data about a route at the given path relative to the current location.

Let's have an example to make it more clear.

```
import { useEffect } from 'react';

import { Route, Routes, useMatch, useNavigate } from 'react-router-dom';

import './App.css';

const Home = () => {
  const navigate = useNavigate();
  return (
    <div>
      <h1>
        Home page
      </h1>
      <button onClick={() => navigate('/profile/John')}>Navigate to Profile </button>
    </div>
  )
}

const Profile = () => {
  const match = useMatch('/profile/:name');
  useEffect(() => {
    console.log("🚀 ~ file: App.js ~ line 23 ~ Profile ~ params", match)
  })
  return (
    <div>
      <h1>Profile Page</h1>
      <p>
        this page is for {match.params.name}
      </p>
    </div>
  )
}

function App() {
  return (
    <div className="App">
```



```
<Routes>
```

```
  <Route path="/" element={<Home />}></Route>
```

```
  <Route index path="/profile/:name" element={<Profile />}></Route>
```

```
</Routes>
```

```
</div>
```

```
);
```

```
}
```

```
export default App;
```

The output of `console.log` will be the below:

```
~ file: App.js ~ line 23 ~ Profile ~ params App.js:20
▼ {params: {...}, pathname: '/profile/John', pathnameBase: '/profile/John', pattern: {...}}
  ► params: {name: 'John'}
    pathname: "/profile/John"
    pathnameBase: "/profile/John"
  ▼ pattern:
    caseSensitive: false
    end: true
    path: "/profile/:name"
    ► [[Prototype]]: Object
    ► [[Prototype]]: Object
```

[< Previous](#)

[next >](#)