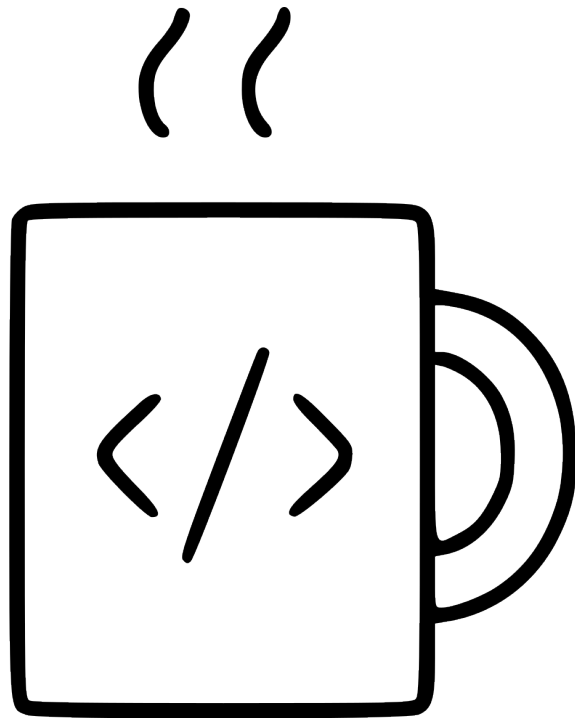


Umberto Emanuele

React – Giorno 5

Gennaio 2023



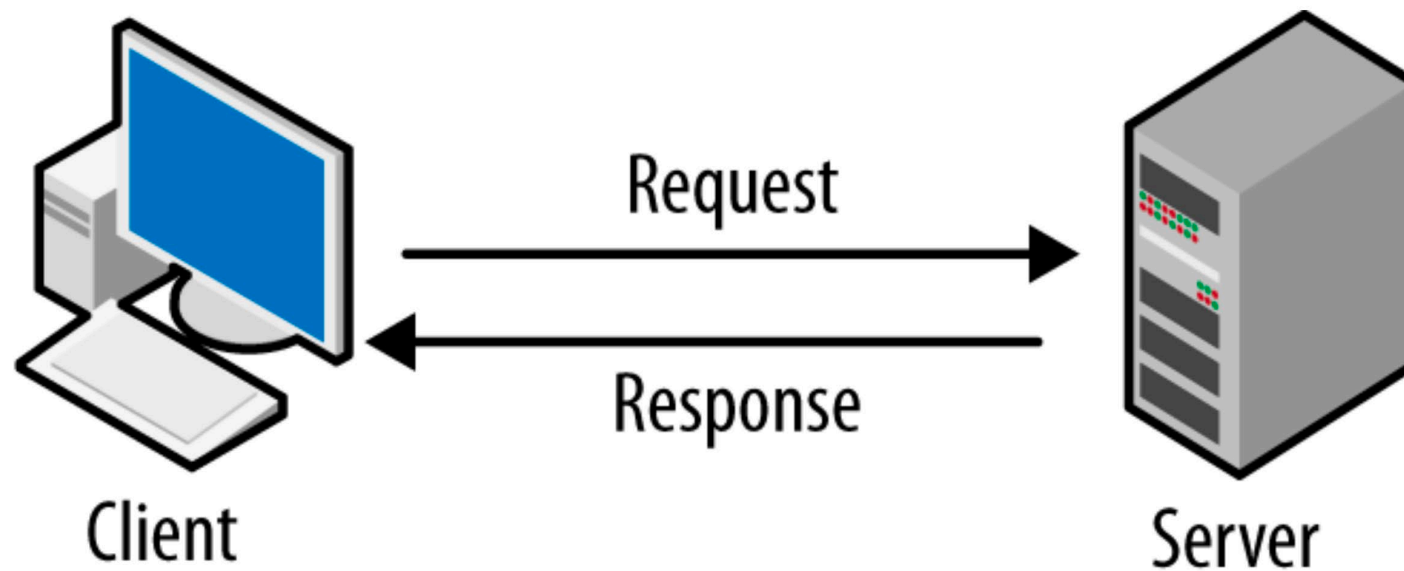


PAUSA

Ci vediamo alle ore 11.25

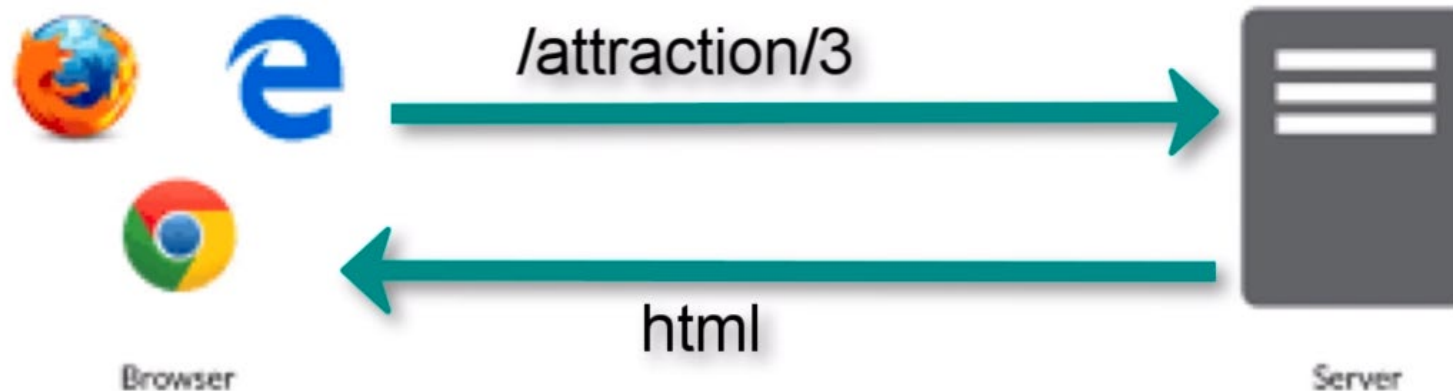
React Router V6

<https://reactrouter.com/>



React Router

React Router ci permette di costruire dei percorsi all'interno della nostra applicazione.



React Router

Le single page application posso funzionare in modi differenti.

1. La SPA carica l'intero contenuto del sito una sola volta. In questo modo quando si naviga nel browser è già tutto disponibile e non c'è bisogno di fare refresh della pagina.
2. La SPA funziona scaricando ogni volta quello che è necessario per fare il render della pagina. Quando l'utente naviga in una nuova pagina, vengono fatte delle richieste Javascript asincrone per ottenere i nuovi dati da presentare.

Oggetto **location** e **location.hash**

localhost:3000/users

▼ General

Request URL: http://localhost:3000/users

Request Method: GET

Status Code: ● 304 Not Modified

Come accedere al valore dopo lo / ?
Utilizzando la proprietà
location.pathname dell'oggetto
window.

localhost:3000/#category

▼ General

Request URL: http://localhost:3000/

Request Method: GET

Status Code: ● 304 Not Modified

Tutto quello che passiamo dopo
#(hash) non viene visto dal **server**
ma solo dal **browser**. Ma ci può
servire per fare il **routing** lato client.

Come accedere al valore
dopo l'hash? Utilizzando
la proprietà
location.hash
dell'oggetto **window**.

React Router v6

Per utilizzare il router è necessario installare tramite NPM react-router

```
npm i -D react-router-dom
```

<https://reactrouter.com/>

```
import ReactDOM from "react-dom/client";
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Layout from "../pages/Layout";
import Home from "../pages/Home";
import Blogs from "../pages/Blogs";
import Contact from "../pages/Contact";
import NoPage from "../pages/NoPage";

export default function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route path="/" element={<Layout />} />
        <Route index element={<Home />} />
        <Route path="blogs" element={<Blogs />} />
        <Route path="contact" element={<Contact />} />
        <Route path="*" element={<NoPage />} />
      </Routes>
    </BrowserRouter>
  );
}

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(<App />);
```

```
import { Outlet, Link } from "react-router-dom";

const Layout = () => {
  return (
    <>
      <nav>
        <ul>
          <li>
            <Link to="/">Home</Link>
          </li>
          <li>
            <Link to="/blogs">Blogs</Link>
          </li>
          <li>
            <Link to="/contact">Contact</Link>
          </li>
        </ul>
      </nav>

      <Outlet />
    </>
  );
};

export default Layout;
```


<BrowserRouter>

<BrowserRouter> memorizza la posizione corrente nella barra degli indirizzi del browser utilizzando URL puliti e naviga utilizzando lo stack della cronologia integrato nel browser.

```
1  import * as React from "react";
2  import * as ReactDOM from "react-dom";
3  import { BrowserRouter } from "react-router-dom";
4
5  ReactDOM.render(
6    <BrowserRouter>
7      { /* The rest of your app goes here */ }
8    </BrowserRouter>,
9    root
10 );
```

<Routes> <Route>

Ogni volta che l'URL cambia, <Routes> esamina tutti i percorsi figlio per trovare la corrispondenza migliore ed esegue il rendering di quel <Route> dell'interfaccia utente.

```
1  <Routes>
2    <Route path="/" element={<Dashboard />}>
3      <Route
4        path="messages"
5        element={<DashboardMessages />}
6      />
7    <Route path="tasks" element={<DashboardTasks />} />
8  </Route>
9  <Route path="about" element={<AboutPage />} />
10 </Routes>
```

<Link>

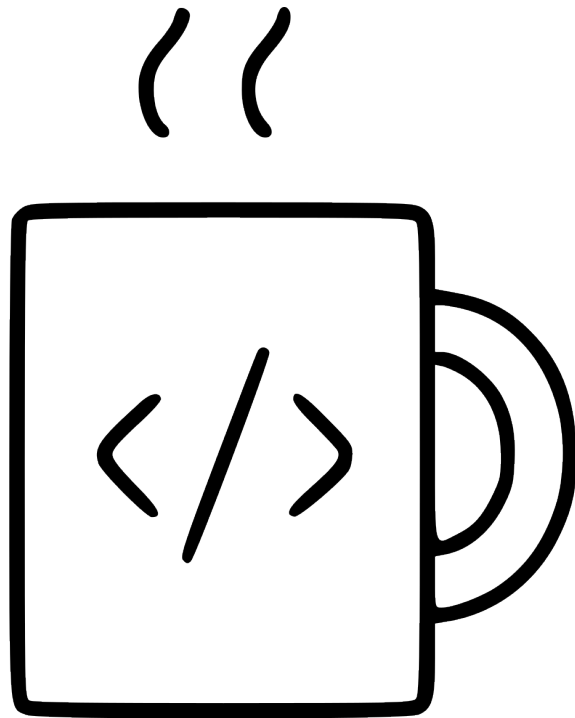
<Link> è un elemento che consente all'utente di passare a un'altra vista senza ricaricare la pagina e senza effettuare nuove chiamate al server

```
1  import * as React from "react";
2  import { Link } from "react-router-dom";
3
4  function UsersIndexPage({ users }) {
5    return (
6      <div>
7        <h1>Users</h1>
8        <ul>
9          {users.map((user) => (
10            <li key={user.id}>
11              <Link to={user.id}>{user.name}</Link>
12            </li>
13          ))}
14        </ul>
15      </div>
16    );
17  }
```

<NavLink>

<NavLink> è un tipo speciale di <Link> che riesce a conoscere se è "attivo" o meno. Ciò è utile quando si crea un menu di navigazione come un breadcrumb o un insieme di schede in cui si desidera mostrare quale di esse è attualmente selezionata. Fornisce inoltre un contesto utile per la tecnologia assistiva come i lettori di schermo.

```
1  import * as React from "react";
2  import { NavLink } from "react-router-dom";
3
4  function NavList() {
5    // This styling will be applied to a <NavLink> when the
6    // route that it links to is currently selected.
7    let activeStyle = {
8      textDecoration: "underline",
9    };
10
11    let activeClassName = "underline";
12
13    return (
14      <nav>
15        <ul>
16          <li>
17            <NavLink
18              to="messages"
19              style={({ isActive }) =>
20                isActive ? activeStyle : undefined
21              }
22            >
23              Messages
24            </NavLink>
25          </li>
26          <li>
27            <NavLink
28              to="tasks"
29              className={({ isActive }) =>
30                isActive ? activeClassName : undefined
31              }
32            >
33              Tasks
34            </NavLink>
35          </li>
```



PAUSA

Ci vediamo alle ore 14.00

redirect

Puoi utilizzare `redirect` per reindirizzare a un altro percorso.

```
1 import { redirect } from "react-router-dom";
2
3 const loader = async () => {
4   const user = await getUser();
5   if (!user) {
6     return redirect("/login");
7   }
8 };
```

useNavigate

L'hook **useNavigate** restituisce una funzione che consente di navigare a livello di codice, ad esempio in uno `useEffect`

```
1 import { useNavigate } from "react-router-dom";
2
3 function useLogoutTimer() {
4   const userIsInactive = useFakeInactiveUser();
5   const navigate = useNavigate();
6
7   useEffect(() => {
8     if (userIsInactive) {
9       fake.logout();
10      navigate("/session-timed-out");
11    }
12  }, [userIsInactive]);
13 }
```


useParams

L'hook `useParams` restituisce un coppia chiave/valore dei parametri dinamici dall'URL corrente che sono stati abbinati da `<Route path>`. Le rotte figlio ereditano tutti i parametri dalle loro rotte padre.

```
1  import * as React from 'react';
2  import { Routes, Route, useParams } from 'react-router-dom';
3
4  function ProfilePage() {
5    // Get the userId param from the URL.
6    let { userId } = useParams();
7    // ...
8  }
9
10 function App() {
11   return (
12     <Routes>
13       <Route path="users">
14         <Route path=":userId" element={<ProfilePage />} />
15         <Route path="me" element={...} />
16       </Route>
17     </Routes>
18   );
19 }
```

React – Router V5 - OLD

Per utilizzare il router è necessario installare tramite NPM react-router

```
npm install react-router-dom
```

```
"dependencies": {  
  "react": "^16.10.2",  
  "react-dom": "^16.10.2",  
  "react-router-dom": "^5.1.2",  
  "react-scripts": "3.2.0"  
},
```

<https://reacttraining.com/react-router/web/guides/quick-start>

```
import React from 'react';
import './App.css';
import { BrowserRouter, Route } from 'react-router-dom';

function App() {
  return (
    <div className="App">
      <header className="App-header">
        <h1>React Router</h1>
      </header>
    </div>
  );
}

function Home() {
  return <h2>Home</h2>;
}

function About() {
  return <h2>About</h2>;
}

function Users() {
  return <h2>Users</h2>;
}

export default App;
```

```
function App() {
  return (
    <div className="App">
      <header className="App-header">
        <h1>React Router</h1>
      </header>
      <BrowserRouter>
        <Route path="/" exact component={Home} />
        <Route path="/about" component={About} />
        <Route path="/users" component={Users} />
      </BrowserRouter>
    </div>
  );
}
```

```
<Route
  exact
  path="/"
  render={() => (
    <div>
      <h1>Welcome</h1>
    </div>
  )}
/>
```

Tramite il **BrowserRouter** vado a collegare le rotte con **Route** indicando il componente che deve essere renderizzato.

Progetto router-app

```
function Users() {  
  return(  
    <React.Fragment>  
      <h2>Users</h2>  
      <Route path="/users/1" component={DettaglioUsers}</Route>  
    </React.Fragment>  
  )  
}  
  
function DettaglioUsers(){  
  return <h3>Dettaglio User</h3>  
}
```

```
function App() {  
  return (  
    <div className="App">  
      <header className="App-header">  
        <h1>React Router</h1>  
      </header>  
      <BrowserRouter>  
        <Route path="/" exact component={Home} />  
        <Route path="/about" component={About} />  
        <Route path="/users" component={Users} />  
      </BrowserRouter>  
    </div>  
  );  
}
```

React Router

Users

Dettaglio User

Rotte annidate

Progetto router-app

```
function Users({match}) {  
  console.log(match);  
  return(  
    <React.Fragment>  
    <h2>Users</h2>  
    <Route path="/users/:id" component={DettaglioUsers} />  
    </React.Fragment>  
  )  
}
```

```
▼ {path: "/users", url: "/users", isExact: false, params: {...}} ⓘ  
  isExact: false  
  ▶ params: {}  
  path: "/users"  
  url: "/users"  
  ▶ __proto__: Object
```

```
function DettaglioUsers({match}){  
  console.log(match);  
  return <h3>Dettaglio User {match.params.id}</h3>  
}
```

```
▼ {path: "/users/:id", url: "/users/3", isExact: true, params: {...}} ⓘ  
  isExact: true  
  ▶ params: {id: "3"}  
  path: "/users/:id"  
  url: "/users/3"  
  ▶ __proto__: Object
```

React Router

Users

Dettaglio User 3

URL parametriche

Progetto router-app

```
function App() {  
  return (  
    <div className="App">  
      <header className="App-header">  
        <h1>React Router</h1>  
      </header>  
      <nav>  
        <a href="/">Home</a>  
        <a href="/about">About</a>  
        <a href="/users">Users</a>  
      </nav>  
      <BrowserRouter>  
        <Route path="/" exact component={Home} />  
        <Route path="/about" component={About} />  
        <Route path="/users" component={Users} />  
      </BrowserRouter>  
    </div>  
  );  
}
```

Inserendo ora dei link possiamo navigare la nostra applicazione. Il problema è che ogni volta fa una chiamata al server.

React Router

[Home](#) [About](#) [Users](#)

Home

Progetto router-app


```
function App() {  
  return (  
    <div className="App">  
      <header className="App-header">  
        <h1>React Router</h1>  
      </header>  
      <nav>  
        <a href="/">Home</a>  
        <a href="/about">About</a>  
        <a href="/users">Users</a>  
      </nav>  
      <BrowserRouter>  
        <Route path="/" exact component={Home} />  
        <Route path="/about" component={About} />  
        <Route path="/users" component={Users} />  
      </BrowserRouter>  
    </div>  
  );  
}
```

```
import { BrowserRouter, Route, Link } from 'react-router-dom';  
  
function App() {  
  return (  
    <div className="App">  
      <header className="App-header">  
        <h1>React Router</h1>  
      </header>  
      <BrowserRouter>  
        <nav>  
          <Link to="/">Home</Link>  
          <Link to="/about">About</Link>  
          <Link to="/users">Users</Link>  
        </nav>  
        <Route path="/" exact component={Home} />  
        <Route path="/about" component={About} />  
        <Route path="/users" component={Users} />  
      </BrowserRouter>  
    </div>  
  );  
}
```

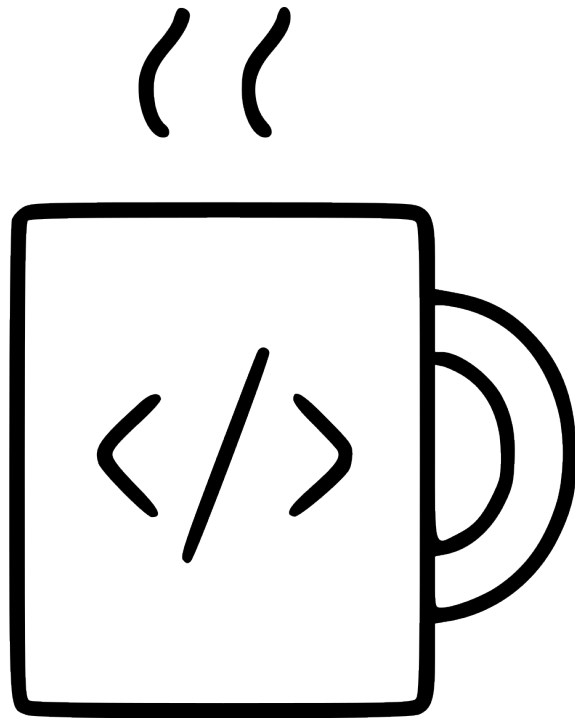
Usando Link al posto
del comune tag html a
evitiamo la chiamata
al server e il
caricamento continuo
di tutte le risorse

```
<NavLink to="/about">About</NavLink>
```

```
<NavLink exact to="/profile">  
  Profile  
</NavLink>
```

```
<NavLink to="/faq" activeClassName="selected">  
  FAQs  
</NavLink>
```

```
<NavLink  
  to="/faq"  
  activeStyle={{  
    fontWeight: "bold",  
    color: "red"  
  }}  
>  
  FAQs  
</NavLink>
```



PAUSA

Ci vediamo alle ore 16.20



shaping the skills of tomorrow

challengenetwork.it

