
REACT
ROUTER

Head of Engineering @ eMoteev

former MTI && ACU @ EPITA

Intro

QUENTIN PRÉ

This is a just a text holder in which you assume that I wrote a lot of interesting things, which obviously I have not.
If you can’t read it easily, you are too far away, get closer to the scene.
If you still have difficulties reading this, raise your hand and ask your question.

RULES

You are encouraged to make mistakes,
You are *forbidden* to make faults.

EFFORT

Do not expect knowledge to fall into your hands.

FOCUS

If you do not intend to follow the lecture, assume consequences for your actions and stay home.

You don’t need your laptop outside of tutorials.

ASK

No question makes you stupid,
Asking no questions though…

note: your question might get answered by another question.

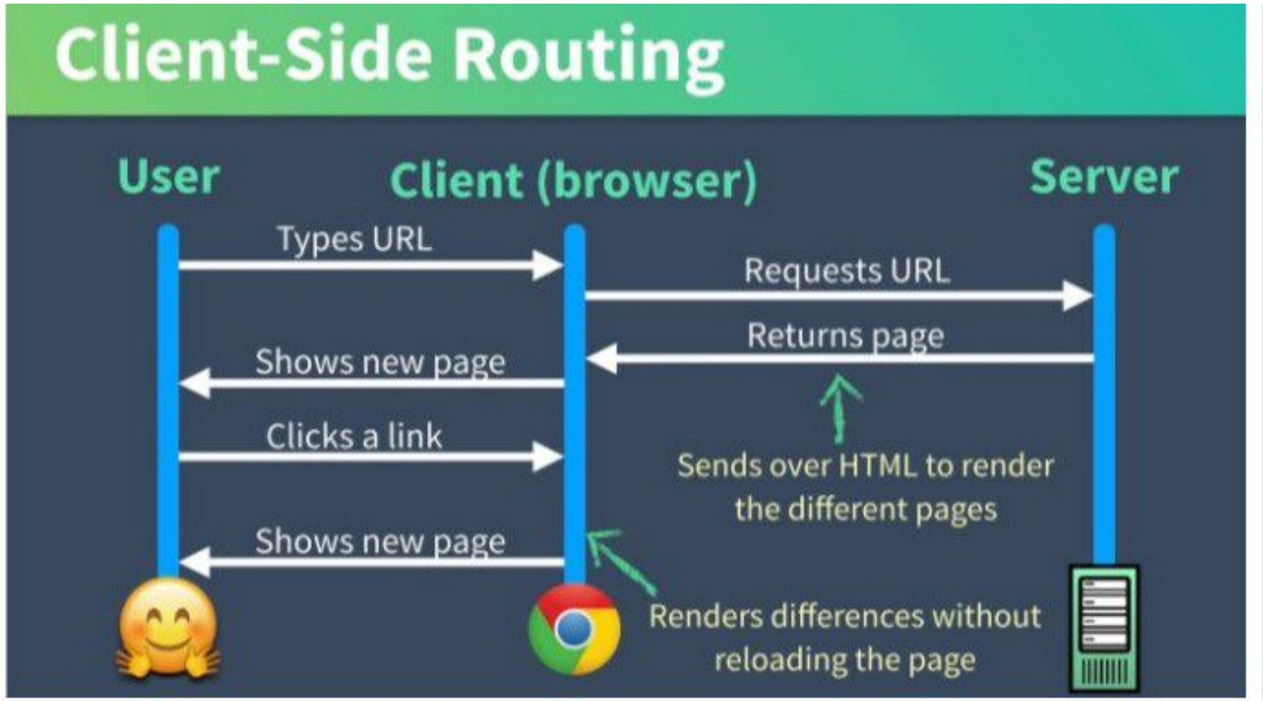
Routing

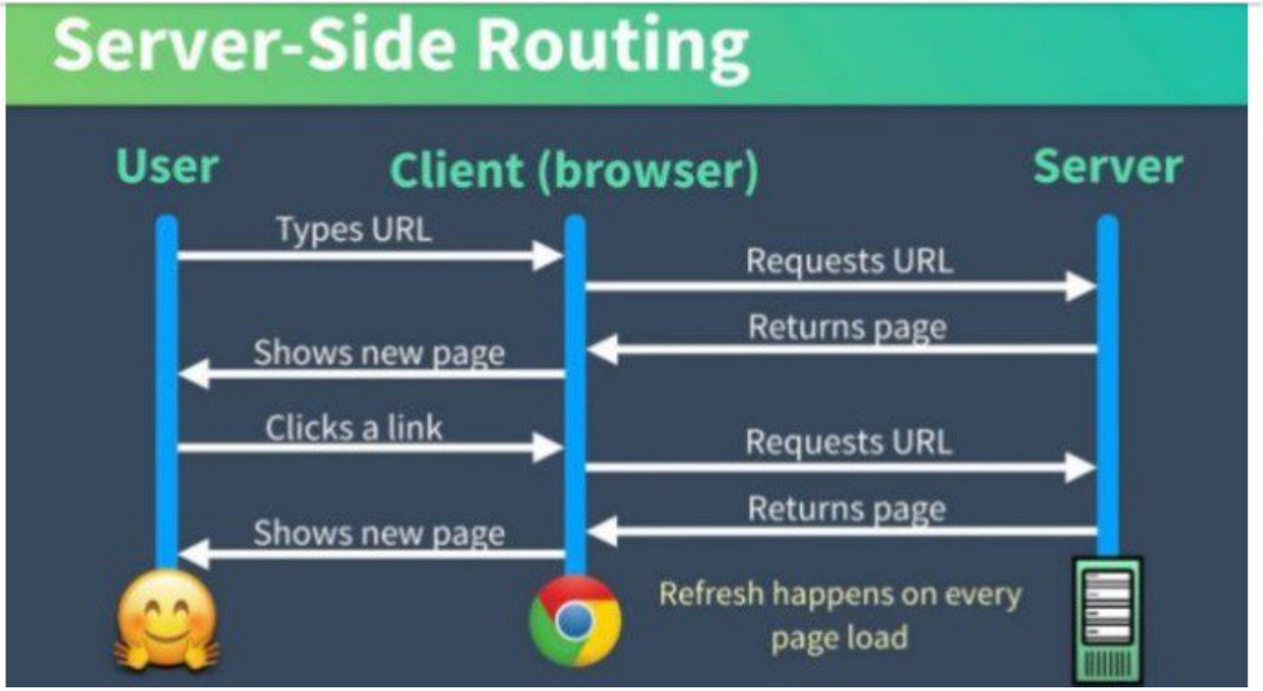
Routing

Definition

- Routing is the ability to render a specific part of the application based on the URL path

- Routing historically happened on the server, but with the rise of **Singe Page Applications (SPA)**, it has become a challenge for the frontend as well


Client-Side Routing
User
Client (browser)
Server
Types URL
Requests URL
Returns page
Shows new page
Clicks a link
Sends over HTML to render the different pages
Shows new page
Renders differences without reloading the page


Server-Side Routing
User
Client (browser)
Server
Types URL
Requests URL
Returns page
Shows new page
Clicks a link
Requests URL
Returns page
Shows new page
Refresh happens on every page load

Client Side Routing with React Router

React Router

Routers

- **BrowserRouter**: a router that uses the HTML5 History API (*pushState*, *replaceState*, *popState*)
  this is the default router
- ***HashRouter**:* a router that uses the hash portion of the URL (after the # in the URL)
  example: *https://toto.com/#profile*
- ***MemoryRouter**:* a router that keeps your location into memory (does not reflect on the address bar)
- ***NativeRouter**:* a router designed for React Native
- ***StaticRouter**:* a router designed for server side rendering

React Router

<Router>

- Common interface for all routers
- Is a React Component
- Acts as a context Provider for sub-components
- *const history = createBrowserHistory();*

```
ReactDOM.render(
  <Router history={history}>
    <App />
  </Router>,
  node
);
```

Route

- A component that will only render if the current path matches its parameters
- **path:** the path to be matched for this route to render
- **exact:** the route will match only if *path* matches exactly *location.pathname*

Path	location.pathname	Exact ?	Matches ?
/one	/one/two	No	Yes
/one	/one/two	Yes	No

- **rendering:** there are three ways for rendering thigh a Route component:
  - render: takes in a function,
  - component: takes in a component,
  - children: sub components in JSX

React Router

```
ReactDOM.render(
  <Router>

    <Route
      path="/home"
      render={() => <div>Home</div>}
    />

    <Route
      path="/dashboard"
      component={Dashboard}
    />

    <Route
      exact
      path="/"
    >
      <IndexPage />
    </Route>

  </Router>,
  node
);
```

React Router

Switch

When wrapping your routes into a <Switch> component, you will only allow the first match to be rendered

In the following example, both *<NavBar>* and *<Content>* are rendered on */*

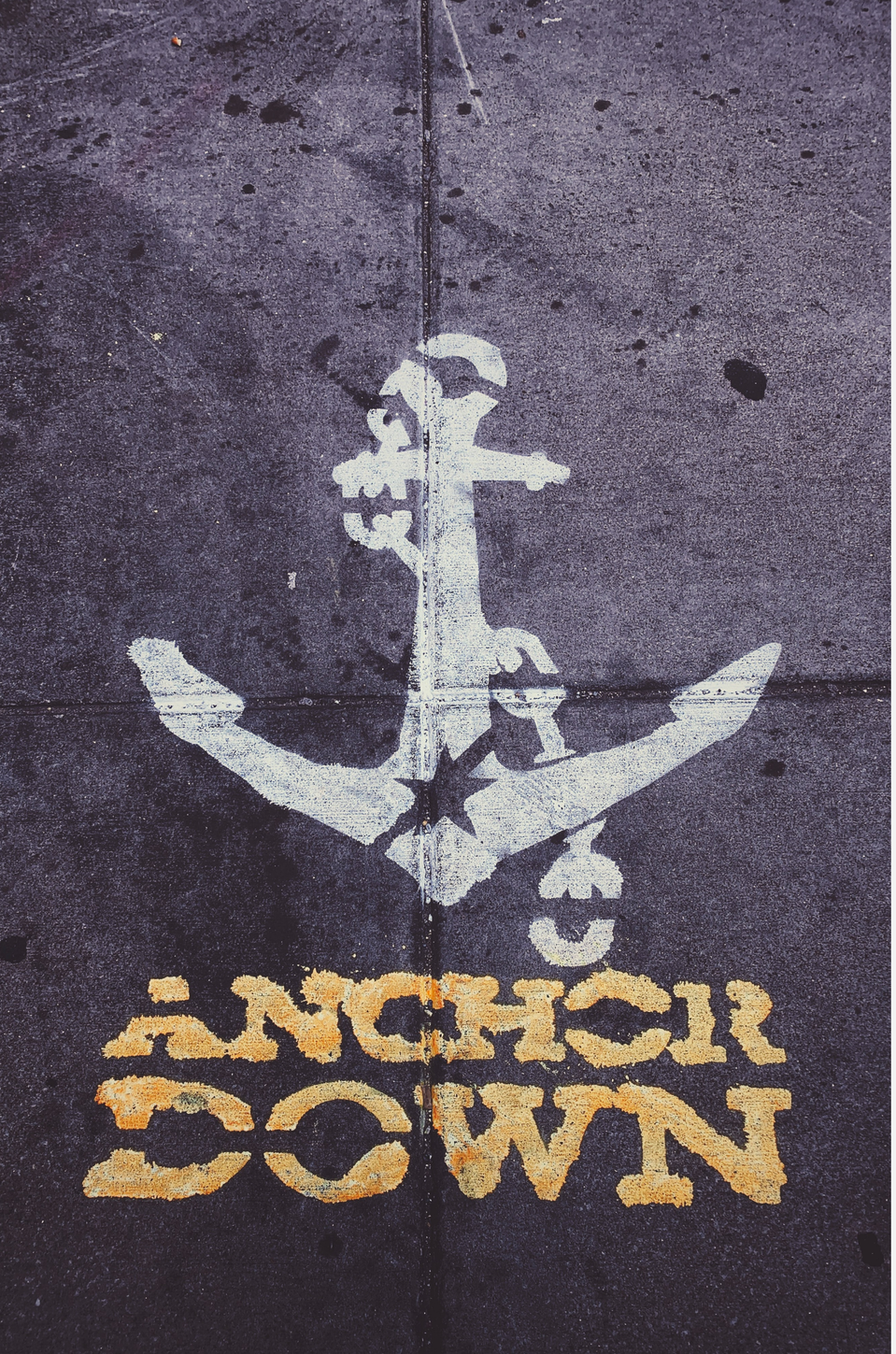
```
ReactDOM.render(
  <Router>
    <Route
      path="/"
      component={NavBar}
    />

    <Route
      exact
      path="/"
      component={Content}
    />
  </Router>,
  node
);
```

In the following example, only *<NavBar>* will be rendered on */*

```
ReactDOM.render(
  <Router>
    <Switch>
      <Route
        path="/"
        component={NavBar}
      />

      <Route
        exact
        path="/"
        component={Content}
      />
    </Switch>
  </Router>,
  node
);
```



React Router

Hooks

- ***useParams():*** this hook gives you a reference to the params in the URL passed in the way of *`/books/**:id**`*

- ***useHistory():*** gives you a reference on the HTML5 history bound to the nearest ***<BrowserRouter />***

- ***useLocation():*** gives you a reference on the current location

```
const MyPage = () => {
    const { user_name } = useParams();
    const  { pathname } = useLocation();
    const history = useHistory();

    return (
      <div>
        Welcome {user_name} !
        you are on {pathname}
        <button
          onClick={() => history.push(
            `/users/${user_name}/dashboard`
          )}
        >
          Go to dashboard !
        </button>
      </div>
    )
}
```

Reads

React Router: Quick Start

React Training

“Always bet on Javascript”

Brendan Eich