

React Routing

Routing

Route Parameters

Query String

Redirect

Not Found Pages

Nested Routing

Grab The Code

https://1drv.ms/f/s!AtGKdbMmNBGdg_FZbzU7IchDO1F17w

As Usual A start and a Finish Project

Two apps

- Vidly
- Router-App

npm install react-router-dom

```
import { Route, Switch, Redirect } from "react-router-dom";
```

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

everything is a component in React Router

2,545,456 Weekly Downloads

Add router to Your app Component

```
import { BrowserRouter as Router, Route, Link } from "react-router-dom"; // at Top
```

Your app Component Should look like this

```
<Router>  
  <div>  
  
  </div>  
</Router>
```

Write a Switch in App.js

```
<Switch>
```

```
  <Route path="/not-found" component={NotFound} />
```

```
  <Route path="/products" component={Products} />
```

```
</Switch>
```

//NotFound is a component.

//which will be rendered when you redirect to

//localhost:3000/not-found

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

```
<Link to="/">Home</Link>
```

Use Link component which is a built in component instead of `Home`

Why ?

Route Props

Route Component is a Router around the component which will be rendered against a url

Route will inject additional props

- History
- Location
- Match

Props

- ▶ `history: {...}`
 - ▶ `location: {...}`
 - ▶ `match: {...}`
-

Router Docs

history: object
children: node

<StaticRouter>
basename: string
location: string
location: object
context: object
children: node

<Switch>
location: object
children: node

context.router

history
history is mutable

location

match
null matches

Philosophy

This guide's purpose is to explain the mental model to have when using React Router. We call it "Dynamic Routing", which is quite different from the "Static Routing" you're probably more familiar with.

Static Routing

If you've used Rails, Express, Ember, Angular etc. you've used static routing. In these frameworks, you declare your routes as part of your app's initialization before any rendering takes place. React Router pre-v4 was also static (mostly). Let's take a look at how to configure routes in express:

```
app.get('/', handleIndex)
app.get('/invoices', handleInvoices)
app.get('/invoices/:id', handleInvoice)
app.get('/invoices/:id/edit', handleInvoiceEdit)

app.listen()
```

Passing Props From Router

Previously

```
<Products sortBy="newest" />
```

Now

```
<Route  
  path="/products"  
  render={props => <Products sortBy="newest"  
    {...props} />}  
 />
```

//Instead of passing component we can render whatever we want

Route Parameters

```
<Route path="/products/:id"  
component={ProductDetails} />
```

Inside ProductDetails Component render method

```
<h1>Product Details -  
{this.props.match.params.id} </h1>
```

//match is passed from route

Optional Parameters

```
<Route path="/posts/:year/:month"  
component={Posts} />
```

```
<Route path="/" exact component={Home} />
```

Year and month are required


If we don't pass any one the route will fail and Home component will be rendered

Optional Parameters

```
<Route path="/posts/:year?/:month?" component={Posts} />
```

Then

```
const Posts = ({ match, location }) => {  
  return (  
    <div>  
      <h1>Posts</h1>  
      Year: {match.params.year} , Month: {match.params.month}  
    </div>  
  );  
};
```



Query String

/posts/2018/09?sortBy=newest&published=1

npm i query-string

```
const Posts = ({ match, location }) => {  
  const result = queryString.parse(location.search);  
  console.log(result);  
  return <div> </div>;  
};
```

Redirects

```
<Route path="/not-found" component={NotFound} />  
<Redirect from="/messages" to="/posts" />
```

If we add

```
<Redirect to="/not-found" />
```

At the end then each unmatched route will be redirected to /not-found

Remember to add redirect at top

```
import { Route, Switch, Redirect } from "react-router-dom";
```

Programmatic Navigation

```
handleSave = () => {  
  // Navigate to /products  
  this.props.history.replace("/products");  
};
```


Nested Routing

```
<Route path="/admin" component={Dashboard} />
```

In Dashboard

```
<SideBar />
```

```
<Route path="/admin/users" component={Users} />
```

```
<Route path="/admin/posts" component={Posts} />
```

//remember to add Route at Top

```
//import { Route } from "react-router-dom";
```

Nested Routing Scannerio



- [Home](#)
- [Products](#)
- [Posts](#)
- [Admin](#)

Admin Dashboard

- [Users](#)
- [Posts](#)

Admin Posts

Comb the code

Comb both sample examples and learn for yourself

https://1drv.ms/f/s!AtGKdbMmNBGdg_FZbzU7IchDO1F17w

It's the same code as given in start of this lecture