



(<https://coursework.vschool.io>)

React Router Nested Routes

27 FEBRUARY 2018

Nested Routes

A common paradigm when creating web applications is to have a "list page" and a "detail page". Consider a product search on Amazon. The results that show up from the search is a **list** of products with some information about each (price, ratings, title, part of the description, etc.). When you click on the product, you're taken to a "detail page" which contains only information (in much more **detail** than before) about that one product.

Think about the example site we've been creating in this post. We have a page about the Services this business offers. Perhaps when the user navigates to the `/services` page, we want to show a list of services to them. However, we also want to allow them to click on one of the services and learn a lot more about that one particular service. Assuming we plan on offering many services (and perhaps adding to the list as we grow as a business), we definitely aren't interested in creating a new React component for every single service we offer. Instead, we want to create a single version of a "Service Detail Page" and just substitute in information about the particular service the user clicks on.

Since we're mostly going to be dealing with JSON data, we'll assume that we can get a list of our services from an API that contains all the information we need to display. Per normal, the database where this information is stored will give each service a unique `_id` property.

Instead of sending them to a new Route (`/lawn-mowing` or `/leaf-raking`) each time, we'll simply append the id of the specific service to the `/services` route. So you'd end up with something like `/services/123` and `/services/234`, where `123` is the `_id` of the "lawn mowing service" and `234` is the `_id` for the "leaf raking service".

This presents a couple of challenges: 1) We need a way to create a `<Route />` that can have a dynamic `path`, and 2) we need a way to reuse a component (`Service`) to display the correct information depending on which `path` they're at. Let's tackle these issues!

Route Params

React Router has a way to let us specify an unknown portion of a Route's `path` using something called a "route param". In the above example, *we don't know what the path will be after* `/services/` (`/services/???`).

Using route parameters, we can say "whatever ends up in this part of the path, let's give it a variable name so we can access it later and figure out how to render the correct thing."

Code time! Let's look at our Services component (which lists the available services, which are stores on `this.state`). Check out the comments along the way for explanation:

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

// This is the component representing an individual service (a
import ServiceDetail from "../ServiceDetail";

class Services extends React.Component {
  constructor() {
    super();
    this.state = {
      services: []
    }
  }

  // imagine this is doing an HTTP call for the data which we
  componentDidMount() {
    this.setState({
      services: [
        {
          name: "Lawn Mowing",
          _id: "123",
          price: 20
        },
        {
          name: "Leaf Raking",
          _id: "234",
          price: 25
        },
        {
          name: "Sprinkler Maintenance",
          _id: "345",
          price: 100
        }
      ]
    })
  }

  render() {
    return (
      <div>
        <div className="services-list">
          {/*
            Create a link for every service. The "to"

```

```

        is directing them to "/services/" + whatever
        of this particular service
    */}
    {this.state.services.map(service => <Link to={`/services/${service._id}`} />)}
  </div>
  { /* This switch is just for the services section
  <Switch>
    { /*
      Notice the :serviceId in the path below.
      This says "whatever shows up here, call it whatever"
      so I can access it later"
    */}
    <Route path={`/${this.props.match.url}/${this.props.match.params.serviceId}`} />
  </Switch>
  </div>
)
}
}

export default Services;

```

Accessing the parameter from the path

Okay, so we told it to save the parameter that shows up in the URL (the `_id` property of the service) as `serviceId`, but how do we access that?

React Router is nice enough to pass to the component it renders (`ServiceDetail`) some props (without us even telling it to!). The important prop we care about is called `match`. `match` is an object that has a `params` property, which is an object containing all the route params we set in the `path` of the `<Route/>` component.

So where above we passed a path of `/services/:serviceId`, inside the `ServiceDetail` component we now have

```
props.match.params.serviceId!
```

How does that help us? Well, typically inside of `ServiceDetail`, we could now do another HTTP request to GET data from our API of that particular item! Using axios, it would look something like this:

```
import React, {Component} from "react";
import axios from "axios";

class ServiceDetail extends Component {
  constructor() {
    super();
    this.state = {
      service: {}
    }
  }

  componentDidMount() {
    axios.get(`http://api.mysite.com/services/${this.props.`
      .then(response => {
        this.setState({
          service: response.data
        });
      });
  }

  render() {
    return (
      <div>
        <h1>{this.state.service.name} - ${this.state.se
      </div>
    )
  }
}

export default ServiceDetail;
```

We now have both a nested view/route (from our Services component we're rendering a dynamic ServiceDetail component), and we've found a way to get our dynamic ServiceDetail

component to render the correct information by using the ID found in `props.match.params.serviceId` to do an API GET request!

Exercise 2: React Wars (<https://coursework.vschool.io/react-wars/>)

(/author/eric/)


Eric Jones (/author/eric/)

Read more posts (/author/eric/) by this author.

📍 Salt Lake

(/author/eric/)

Share this post

 ([https://twitter.com/text=React%20Router%20router-nested-views-2,](https://twitter.com/text=React%20Router%20router-nested-views-2)

Search for another post

SEARCH!

(/fizz-buzz/)

READ THIS NEXT

Fizzbuzz

Write a function that returns an array containing the numbers 1 through 100 with the following stipulations: If the...

(/array-shuffler/)

YOU MIGHT ENJOY

Array Shuffler

Write a function that takes an array as a parameter and returns a new array with the contents randomly...