**NOTE 3D**

**MANAGING APP LOCATION WITH REACT ROUTER**

Q: What’s Single Page application?

A: Meaning that every time a user access a page, the browser doesn’t need to go back to the server to fetch the pages. Instead, JS handles the transition => Only 1 initial page from the server

Q: So how do Single App pages do that?

A: It either download everything at once , or only renders just what the user needs

Q: How does React Router help?

A: It turns React projects into single-page app by providing specialized components that manage the creation of links, manage the app’s URL, provide transitions when navigating between different URL locations and so much more

Q: What is this syntax

{this.state.screen === 'list' && (

<ListContacts onDeleteContact= {this.removeContact} contacts={this.state.contacts}/>

)}

{this.state.screen === 'create' &&(

<CreateContact />

)}

A: just JSX if statements. If condition is “this.state.screen=== ‘list”, then execute the code.

It’s called **short- circuit evaluation**

Q: Why use React Router?

A: So I can create urls for each page of the app

npm install --save react-router-dom (DOM is for web page. Native is for React Native)

**BrowserRouter**

Q: What’s that?

A: It listens to the changes in URL and it makes sure that the correct screen correlates to that new url shows up

\_ it has a “history” object that listen to changes in the URL and makes sure that my app is aware of those changes

\_ So I need to import it into index.js and then wrap it around the App component (wrap my whole app around it):

ex:

import { BrowserRouter} from 'react-router-dom'

<BrowserRouter><App /></BrowserRouter>

**Link Component**

Q: What’s that?

A: For users to navigate your app.

import { Link } from 'react-router-dom'

When user clicks the link, it talks to the browser and tells it to update the url

Like a href (creating a link)

So this piece of code below:

<a

href='#create'

onClick={this.props.onNavigate}

className='add-contact'

>Add Contact</Link>

</div>

becomes:

<Link

to='/create'

className='add-contact'

>Add Contact</a>

* Notice “to” instead of href. And now it has an actual link
* a -> Link
* No need for onClick cuz React Router handles that

I can also pass state to the new Link:

Ex:

<Link to={{

pathname: '/courses',

search: '?sort=name',

hash: '#the-hash',

state: { fromDashboard: true }

}}>

Courses

</Link>

**ROUTE COMPONENT**

Q: What does that do?

A: It takes a path, and see if it matches the url or not. If it does, return a UI. If not, it does nothing.

Ex:

<Route

path= “/create”

render={ui}

/>

Q: Look at this code, what do I notice?

<div className = "app">

<Route exact path='/' render={() => (

<ListContacts

onDeleteContact={this.removeContact}

contacts={this.state.contacts}

/>

)}/>

<Route path ="/create" component={CreateContact} />

A: render=()

Exact path is so that if the path isn’t exact, it won’t render the right component. Because “/create” has “/” too

It’s basically just 2 if statements, but written in the React Router way

Notice that 1 of them renders a custom UI, the other renders a component. But for the second way I don’t have to say “render” anymore; instead uses “component”

And no more onNavigate

\_Also, a CSS error can make things not show up

ex:

div className = 'list-contacts'>

<div className= 'list-contacts-top'>

<input

className= 'search-contacts'

type= 'text'

placeholder= 'Search contacts'

value={query}

onChange={(event) => this.updateQuery(event.target.value)}

/>

<Link

to='/create'

className='add-contact'

>Add Contact</Link>

</div>

I used to have the one in yellow as “List-contacts” and one of my icons wouldn’t show. Still don’t know why.

Q: How to create a form?

A:

<div className= "create-contact-details">

<input type="text" name="name" placeholder="Name"/>

<input type="text" name="name" placeholder="email"/>

</div>

Q: What’s the “form-serialize” package?

A: To serialize and put into the query

Ex:

handleSubmit = (e) =>{

e.preventDefault()

const values = serializeForm(e.target, {hash: true})

console.log(values)

}

* So it’ll prevent the form from submitting
* But it will serialize the event into an object and hash true to submit it

Q; What’s a hash object?

A:

Q: Ref and DOM?

A:

Q: Where js the variable being kept?

A: In the memory

Q: How to add an item into an array?

A: Push or Concat. Below:

createContact(contact){

ContactAPI.create(contact).then(contact =>{

this.setState(state =>({

contacts: state.contacts.concat([ contact])

}))

})

}

// this adds the concated item (contact) to the contacts array

// also notice I use state.contacts, not this.state.contacts

Q: I have a key error!

A: Then look at this

<div className= 'create-contact-details'>

<input type='text' name='name' placeholder='Name'/>

<input type='text' name='email' placeholder='Email'/>

<button> Add Contact </button>

(Earlier, I gave them the same “name”, so of course they’re taking the same key value!!!)

\_History.push just push us back to the original url (the home page)