**Project note**

Q: How to use ‘for Each’?

A: mySimpleArray = ['a', 1, 'etc'];

mySimpleArray.forEach(function(val) {

console.log(val);

});

* Used over arrays

Q: for-in:

A: Used over objects

his can be error prone because for-in does not loop from 0 to length - 1 but over all the present keys in the object and its prototype chain.

* So:

myObj = {'firstName':'Ada','lastName':'Lovelace'};

for (var key in myObj) {

if (myObj.hasOwnProperty(key)) {

console.log(myObj[key]);

}

}

\_For new repo on github, use git remote add origin

Q: Sometimes you create a remote repo and a local repo. Then when you try to merge the 2 together (pull), you receive this message “fatal: refusing to merge unrelated histories

“. What do you do?

A: git pull origin branchname --allow-unrelated-histories

Q: componentDidMount(){

BooksAPI.getAll().then((books) =>{

this.state({books})

})

}

A: getAll goes into the BooksAPI where it’s defined, and then it goes through all the promises and saves into this.state. books: books is simplified to just “books”

Q: So when do we use =() =>{} on the method and when do we just use method()?

A: They’re basically the same thing if we consider what’s written on the render() method:

1. Properly initialized syntax:
2. class LoggingButton extends React.Component {
3. // This syntax ensures `this` is bound within handleClick.
4. // Warning: this is \*experimental\* syntax.
5. handleClick = () => {
6. console.log('this is:', this);
7. }
8. render() {
9. return (
10. <button onClick={this.handleClick}>
11. Click me
12. </button>
13. );
14. }
15. }

2) Non-properly initialized syntax:

class LoggingButton extends React.Component {

handleClick() {

console.log('this is:', this);

}

render() {

// This syntax ensures `this` is bound within handleClick

return (

<button onClick={(e) => this.handleClick(e)}>

Click me

</button>

);

}

}

* They are the same, but preferably use ()=>{}. Just different syntaxes

\_If you write it like this bookAPI.update(book, shelf).then((book, shelf)=>{

It’s like you’re redefining book and shelf.

So instead, do this: bookAPI.update(book, shelf).then(()=>{

\_Always pass an object into setState

\_componentDidMount(){

BooksAPI.getAll().then((books) =>{

this.setState({books})

console.log(books)

})

}

* Notice no comma between this.setState() and console.log

Q: What does this do?

UpdateBooks = (book, shelf) =>{

BooksAPI.update(book, shelf).then(()=>{

book.shelf = shelf;

this.setState({

books: this.state.filter(b => b.id !== book.id).concat([ book ])

})

})

}

A: First I make the API call.

Then API call returns a promise, which is resolved using .then(…)

Inside .then we have () =>{…}

Now the method inside is executed.

book is a new variable, and the same with ‘shelf’. Book.shelf is set to ‘shelf’.

Now in the books array, there already exist an old variable book which contains the same id of the book the user is selecting, but this one has no shelf

Then when you run the .filter array, it returns a new array WITHOUT the old variable.

Then now you concat it (as opposed to ‘push’, to create a new array) that item to the new book array.

We return the new array cuz we need to treat state as immutable

Another way is:

updateBook = (book, shelf) => {

const books = [...this.state.books];

books.forEach(b => {

if(b.id === book.id) {

b.shelf = shelf;

}

});

\_When I use React Router, I’ve got to wrap it in the index.js file, wrapping it around <App> like this:

<BrowserRouter><App /></BrowserRouter>

\_Also, when import file with 1 component/ function in, use “import filename from direction”

ex: import BookShelves from './BookShelves'

\_But with files with multiple components/ functions, use “import \* as name from direction”

ex: import \* as BooksAPI from './BooksAPI'

Your function has ato return a view

For example, this doesn’t work:

{Shelves.map((shelf) =>{

<h2 className="bookshelf-title">{shelf.title}</h2>

})}

* Need to return 1 view, like below:

{Shelves.map((shelf) =>{

return(

<h2 className="bookshelf-title">{shelf.title}</h2>

)

})}

\_If I wanna pass something in so that it’ll show up, pass it like this:

<div className="book-list-item”> {book1.title</div>

\_Don’t pass things into a plain div element like this (it might givde you an error

<div props props>

\_ Whenever you see this error: “Objects are not valid as a React child (found: object with keys {title, subtitle, authors, publisher, publishedDate, description, industryIdentifiers, readingModes, pageCount, printType, categories, averageRating, ratingsCount, maturityRating, allowAnonLogging, contentVersion, panelizationSummary, imageLinks, language, previewLink, infoLink, canonicalVolumeLink, id, shelf}). If you meant to render a collection of children, use an array instead or wrap the object using createFragment(object) from the React add-ons. Check the render method of BookShelf.

lVolumeLink, id, shelf}). If you meant to render a collection of children, use an array instead or wrap the object using createFragment(object) from the React add-ons. Check the render method of BookShelf.”

* It could be that one of your prop is an object

\_If I want to display something down the prop, I have to call {this.props.children} on the child component

Ex: Passing {shelf.title} from BookShelves to BookShelf

BookShelves:

<div>

{Shelves.map((shelf) =>{

return(

<BookShelf

className="bookshelf-content"

shelf={shelf}

title={shelf.title}

key={shelf.id}

isRightBook={shelf.bookOnShelf}

books={books}

updateBookShelf={updateBookShelf}

>{shelf.title}

</BookShelf>

)

})}

</div>

And then BookShelf:

eturn(

<div className="bookshelf-books">

<div className="bookshelf-title">{this.props.children}</div>

<ol className="books-grid">

<li>

\_NEVER forget “export default component”. This fucks me up so many times

\_If I do .map((something , I can pass that something as a prop too!

\_Annoying thing about React (which will be fixed in Redux) is that I have to keep passing things down as props from layer to layer

Q: How to join 2 strings together, separated by commas?

A: .join(“ , “)

Q: how to use teritiary condition?

A: (book.authors.length >1) ? book.authors.join(" , "): book.authors

\_Need to put in weight and height for the images

\_For tag selectors and options, like below:

<div className="book-shelf-changer">

<select value ={book.shelf} onChange={(e) => {updateBookShelf(book, e.target.value)}} >

<option value="none" disabled>Move to...</option>

<option value="currentlyReading">Currently Reading</option>

<option value="wantToRead">Want to Read</option>

<option value="read">Read</option>

<option value="none">None</option>

</select>

</div>

//set book.shelf so that the first tag selected will be that shelf

//Also, notice it’s e.target.value, not just “e”. I wrote “e” earlier and it didn’t work!

Q: Don’t use {this.props.children}. Why?

A:

Q: Which line is nonsensical?

<div>

{Shelves.map((shelf) =>{

return(

<BookShelf

className="bookshelf-content"

shelf={shelf}

shelfTitle={shelf.title}

key={shelf.id}

isRightBook={shelf.bookOnShelf}

books={books}

updateBookShelf={updateBookShelf}

>{shelf.title}

</BookShelf>

)

})}

</div>

A: line in yellow. It’s not necessary. React will just ignore it. What matters is where you pass it at the children’s component using props (over this.props.children)

\_import escapeRegExp from 'escape-string-regexp'

Q: What’s wrong with this code, and how to properly write it?

<Link className="open-search" to='/search'>

<a>Add a book</a>

</Link>

A:

What’s wrong? <a> is the same thing as Link. So of course when you put the <a> within the <Link>, an error will pop up

Correct way:

<div className="open-search">

<Link to="/search">Add a book</Link>

</div>

//Why it work? Cuz Link is the same as <a>!!

Q: Where should I put that code?

A: I need to put it in the innermost component, because in this case event propagates up. If I put it into App.js, it’s gonna cascade down to all components, including Search. Can’t put it in Search because you don’t want the button to show up on the Search page.

But it in the Bookshelf component

Q: Why use promises?

A: To catch error

Q: How to catch error?

A:

Search = (query) => {

this.setState({query: query.trim() })

BooksAPI.search(query.trim(), 50).then( (bookResults) =>{

// if there's no book matching the query

if(!bookResults || bookResults.error){

console.log('no books matching the query or error')

this.setState({bookResult: []})

} else{

bookResults.sort(sortBy('title'))

this.setState({bookResult: [bookResults]})

console.log('Found book(s)')

}

})

}

Q: TypeError: .map isn’t a function. How to solve?

A: It means that it’s not an array

Q: How to write promise?

A: First, write a .then(() => 2 ifs, and a catch

Ex:

updateBookShelf = (book, shelf) =>{

//update state and BooksAPI

let myLibraryBooks = this.state.books

// why do we need shelves here?

BooksAPI.update(book, shelf).then((shelves) => {

//find the book being modified in my library if it exists

let modifiedBook = myLibraryBooks.find((aBook)=> {

return aBook.id === book.id

})

if (!modifiedBook){

// if book isn't in my library already append with the new shelf

book.shelf = shelf

myLibraryBooks.push(book)

} else {

//Update modifiedBook so that the book in question has its shelf updated.

modifiedBook.shelf = shelf

}

//update state of myLibraryBooks

this.setState( (state) => ({

books: myLibraryBooks

})

)

}).catch((error) => {

//catch any errors in console

console.error(error)

})

}

Q: Atom—How to autoindent?

A: Edit -> Lines -> AutoIndent

Q: How to split screen?

A: View-> Panes -> Split Left or Right

Q: Why do I have to refer to a function outside of render() as “this.nameOfFunction”, but I don’t have to when I define that function within Render()?

A:

\_When you define a function inside of another, you're actually creating a variable inside of that function, so you can reference it directly.

Ex:

function outerFunc () {

const innerFunc = () => {

console.log("This function is a local variable within outerFunc");

}

innerFunc();

}

\_If you create a function outside of the `render` method, that function will be a property of your component class. You must use `this` to reference to any property in your object. The context of ‘this’ is the Component

Ex:

class Search extends React.Component {

objFunc = () => {

console.log("objFunc is part of our Search object.");

}

render () {

this.objFunc();

}

}

Q: Arrow function. When do I use {}, when do I use? ()?

A: 1) If I need to do more things in my function aside from ‘return’, I need to use {}. ‘return’ goes with {}

2) if I only need to return, use ().

Ex:

\_This doesn’t return anything:

{bookResult.map((book) => { <JSX code> })}

\_This does:

{bookResult.map((book) => { return (<JSX code>) })}

\_Also this does:

{bookResult.map((book) => (<JSX code>))}

Q: Most effective way to debug?

A: Console.log out that element

Search = (query) => {

this.setState({query: query.trim() })

BooksAPI.search(query.trim(), 50).then( (bookResults) =>{

// if there's no book matching the query

if(!bookResults || bookResults.error){

console.log('no books matching the query or error')

this.setState({bookResult: []})

} else{

bookResults.sort(sortBy('title'))

this.setState({bookResult: bookResults})

console.log('Found book(s)')

console.log(bookResults)

}

}).catch((error) => {

//catch any errors in console

console.error(error)

})

}

//see🡺 I console.logged out bookResults

Q: When should I use PropTypes? And how to import?

A: \_Import: import PropTypes from 'prop-types'

\_I should use it to restrict what types of data is passed into props so it’s easier to debug:

Like this:

const {books, updateBookShelf, wholeLibrary} = this.props

Now:

static propTypes ={

updateBookShelf: PropTypes.func.isRequired,

wholeLibrary: PropTypes.array.isRequired

}

Also there’s .string

Q: So what type of array is this (bookResult)?

state = {

query: '',

bookResult: []

}

this.setState({bookResult: [bookResults]})

A: 3 dimensional arrays

Q: Should I use .map over the <ol> or <ul>. And what’s the difference between the 2?

A:

<ol> defines an unordered list, creating that list

<ul> generates each line of the list

* So I have to map the array over each line of the list, not to the whole list itself:

<ol className="books-grid">

{bookResult.map((book) =>{

return (<li key={book.id} className="book">

<Book

key={book.id}

books={books}

updateBookShelf={updateBookShelf}

bookTitle={book.title}

book={book}>

</Book>

</li>

)

})

}

</ol>

Q: Defining key?

A:

So can’t write it like this: key= “book.id”

Wrong! Because there would be so many keys like that

But, write it like this: key={book.id} . {} denotes Javascript