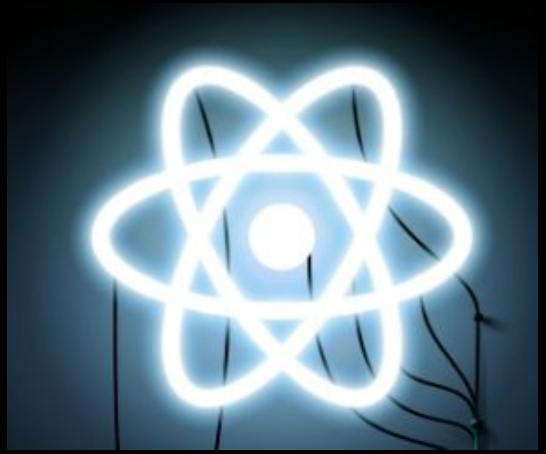
Lecture 4.0

Passing Data Between Components



Topics

- Defining Basic Routes
- Defining Multiple Routes
- Route Parameters
- Navigating with History

How to pass data between components?

Because of React's one-way data flow, it often can be tricky to see how data can flow from one component to another.

Data sometimes needs to be able to move from children to parent, parent to children, or between siblings.

```
App
 |
 |__ InputBar
 |
 |__ ToDoList
       |
       |__ToDoItem
```

Parent to Child - Use Prop

This is the easiest direction in React to transfer data. To access to data in my parent component that I need my child component to have access to, I can pass it as a prop to the child when I instantiate it within the parent.

```
class App extends React.Component {
    render() {
        const listName = "toDoList";

        return (
            <div>
                <InputBar/>
                <ToDoList listNameFromParent={listName}/>
            </div>
        );
    }
}
```

Now in the ToDoList component, use this.props.listNameFromParent to access to that data

Child to Parent - Use a callback and state

This one is a bit trickier. If I have data in my child that my parent needs access to, I can do the following:

1. Define a callback in my parent which takes the data I need in as a parameter.
2. Pass that callback as a prop to the child (see above).
3. Call the callback using this.props.[callback] in the child (insert your own name where it says [callback] of course), and pass in the data as the argument.

Child to Parent - Use a callback and state

Here's what that might look like if I had data in ToDoItem (Parent) that I need to access in ToDoList (Child):

```
class ToDoList extends React.Component {

    myCallback = (dataFromChild) => {
        //...we will use the dataFromChild here...
    }

    render() {
        return (
            <div>
                <ToDoItem callbackFromParent={this.myCallback}/>
            </div>
        );
    }
}
```

Child to Parent - Use a callback and state

Now from within ToDoItem (Child) we can pass something to callbackFromParent:

```
class ToDoItem extends React.Component{

    someFn = () => {
        const listInfo = "something important to do";

        this.props.callbackFromParent(listInfo);
    }
    render() {
        //...
    }
};
```

ToDoList (Parent) will now be able to use listInfo within it's myCallback function!

But wait...

But what if I want to use listInfo in a different function within ToDoList (Parent), not just in myCallback? With this implementation, we would only have access as a parameter passed into that one specific method.

Easy: set this parameter as a state within ToDoList. You can almost think of it as creating a variable within the scope of ToDoList that all the methods within that component can access. In that case my code defining ToDoList might look something like:

```
class ToDoList extends React.Component {
    constructor(props) {
        super(props);
        this.state = {
            listDataFromChild: null
        };
    }
    myCallback = (dataFromChild) => {
        this.setState({ listDataFromChild: dataFromChild });
    }
    otherFn = () => {
        //...we have access to this.state.listDataFromChild...
    }
    render() {
        return (
            <>
                <ToDoItem callbackFromParent={this.myCallback}/>
                <SiblingItem data={this.state.listDataFromChild} />
            </>
        );
    }
}
```

1. Between Siblings - Combine previous

Not surprisingly, to pass data between siblings, you have to use the parent as an intermediary.

1. First pass the data from the child to the parent, as an argument into a callback from the parent.
2. Set this incoming parameter as a state on the parent component,
3. Then pass it as a prop to the other child. The sibling can then use the data as a prop.

Passing data between React components can be a little tricky at first (without using Redux that is), but once you practice these three techniques you'll be able to pass data between whichever components you'd like.

Video: Passing Data between components

https://youtu.be/AnRDdEz1FJc