CODERSINHOODS

React useEffect, fetch, custom hooks

Component Lifecycle

What do you think this means?

What are some moments in a component’s lifecycle?

Component Lifecycle

Birth

Everything in between

Death

Component Lifecycle

~~Birth~~

Mounting

~~Everything in between~~

Updating

~~Death~~

Unmounting

CODERSINHOODS

Component Lifecycle

useEffect covers all of these

A function which takes a callback

The callback is called whenever a *dependency* changes

Dependencies are specified in an array

useEffect - updating

```
const [counter, setCounter] = useState(0);

useEffect(() => {
        console.log(“counter has been updated!”);
}, [counter]);
```

useEffect - updating

```
const [counter, setCounter] = useState(0);

useEffect(() => {
        console.log(“counter has been updated!”);
}, [counter]);
```

Counter is a dependency

The callback is called every time it changes

useEffect - updating

To the code editor

useEffect - mounting

```
useEffect(() => {
        console.log(“component has mounted!”);
}, []);
```

useEffect - mounting

```
useEffect(() => {
        console.log(“component has mounted!”);
}, []);
```

Dependency array is empty

Callback is called whenever a dependency changes

No dependencies == no calls to callback?

Not quite, there’s 1 initial call

useEffect - initial API call

```
useEffect(async () => {
    const birds = await API.getBirds();
    setBirds(birds);
}, []);

if (birds.length === 0) return <div>LOADING BIRDS</div>

return <div>
    <BirdsList birds={birds} />
</div>
```

useEffect - mounting

To the code editor

useEffect - unmounting

Why do we care about unmounting?

useEffect - unmounting

Async!

Updating state on a component which isn’t there is bad

Before unmounting, cancel/handle async actions

useEffect - unmounting

The callback passed to useEffect can return another function

This gets called before unmounting

```
useEffect(() => {
    const timeoutId = setTimeout(doStuff, 1000);

    return () => {
        clearTimeout(timeoutId);
    }
}, []);
```

useEffect - unmounting

To the code editor

useEffect

Any questions on useEffect?

Custom hooks

Hooks, like components, should be reusable

Forms are a good example

Custom hooks are functions which start with **use**

Custom hooks

```
const [formData, setFormData] = useState({
      firstName: “”,
      lastName: “”
});

const handleChange = event => {
      setFormData({
      ...formData,
      [event.target.name]: event.target.value
      });
}

const handleSubmit = event => {
      event.preventDefault();
      API.postData(formData);
}
```

Custom hooks

```
const [formData, handleChange, handleSubmit] = useForm({
     firstName: “”,
     lastName: “”
}, () => {
     API.postData(formData);
});
```

Custom hooks

To the code editor

Custom hooks

https://usehooks.com/

Any questions on custom hooks?