

Hooks, Component Composition, and PropTypes



Cory House

PRINCIPAL CONSULTANT

@housecor reactjsconsulting.com



Agenda



Hooks

Component composition

- Controller views

Prop validation via PropTypes



Hooks



Why Hooks?

- Use functions for (nearly) everything
- Avoids ``this`` confusion
- Use the same React concepts
- Better model the way React works
- Easier to work with
- Share logic between components



Hooks



useState – Local state

useEffect – Side effects

useState

```
import React, { useState } from "react";
```

```
function Example() {  
  const [email, setEmail] = useState("");
```

```
  return (  
    <input  
      type="text"  
      value={email}  
      onChange={event => setEmail(event.target.value)}  
    />  
  );  
}
```



useEffect

```
import React, { useState, useEffect } from "react";

function Example() {
  const [count, setCount] = useState(0);

  useEffect(() => {
    document.title = `You clicked ${count} times`;
  });

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  );
}
```



useEffect

```
import React, { useState, useEffect } from "react";

function Example() {
  const [count, setCount] = useState(0);

  useEffect(() => {
    document.title = `You clicked ${count} times`;
  }, [count]);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  );
}
```



useEffect
componentDidMount +
componentDidUpdate +
componentWillUnmount

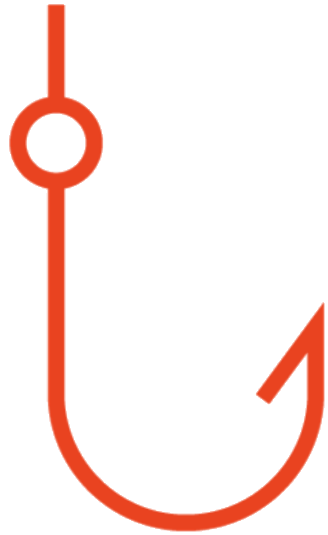


Hooks

Share logic between components



Rules of Hooks



Only call in

- React function components
- Custom Hooks

Hooks must be declared at the top level

- Don't wrap in if statements, loops, funcs
- Why? React tracks Hook call order



Demo



Hooks



Composing Components

```
function ProfilePic(props) {  
  return <img src={"/images/" + props.username + ".png"} />;  
}
```

```
function ProfileLink(props) {  
  return <a href={"/profiles/" + props.username}>{props.username}</a>;  
}
```

```
function Avatar(props) {  
  return (  
    <div>  
      <ProfilePic username={props.username} />  
      <ProfileLink username={props.username} />  
    </div>  
  );  
}
```



Controller Views

**Top level
component**

**Sets props on
children**

**Interacts with
stores**



Controller Views: Related Terms

Smart	Dumb
Container	Presentation



Demo



Controller Views



PropTypes

```
CoursePage.propTypes = {  
  author: PropTypes.object.isRequired,  
  onSave: PropTypes.func.isRequired,  
  validate: PropTypes.func.isRequired,  
  errors: PropTypes.object,  
  hasErrors: PropTypes.func.isRequired  
};
```



PropTypes

optionalArray: PropTypes.array,
optionalBool: PropTypes.bool,
optionalFunc: PropTypes.func,
optionalNumber: PropTypes.number,
optionalObject: PropTypes.object,
optionalString: PropTypes.string



Development vs Production

PropTypes only run in the development version of React.

The minified React version is for production.



Default Props

```
CoursePage.defaultProps = {  
  errors: {}  
};
```



Demo



PropTypes



Summary



Hooks

- Manage state - useState, useReducer
- Run code at different times in funcs
- Share logic

Controller views

- “Smart” components
- Pass data and funcs down via props
- Container/Presentation
- Smart/Dumb

PropTypes

- Document and validate props

Next up: React Router

