

Setting foot on React Hooks

Polidea 



Table Content

- Why is there React Hooks feature?
- What are React Hooks?
- How to implement React Hooks APIs?
- Hook the right way



Why?



Giant Components

```
import React from 'react'
import { Card, Row, Input, Text } from './components'
import ThemeContext from './ThemeContext'

export default class Greetings extends React.Component {
  constructor(props) {
    super(props)
    this.state = {
      name: '',
      surname: '',
      width: 100
    }
  }

  componentDidMount() {
    // ...
  }

  componentDidUpdate() {
    // ...
  }

  componentWillUnmount() {
    // ...
  }

  render() {
    const { name, surname, width } = this.state
    return (
      <Card theme={theme}>
        <Row label="Name">
          <Input value={name} onChange={this.handleNameChange} />
        </Row>
        <Row label="Surname">
          <Input value={surname} onChange={this.handleSurnameChange} />
        </Row>
        <Row label="Width">
          <Text>{width}</Text>
        </Row>
      </Card>
    )
  }
}
```

```
import React, { useState, useContext, useEffect } from 'react'
import { Card, Row, Input, Text } from './components'
import ThemeContext from './ThemeContext'

export default function Greetings(props) {
  return (
    <Card theme={theme}>
      <Row label="Name">
        <Input {...name} />
      </Row>
      <Row label="Surname">
        <Input {...surname} />
      </Row>
      <Row label="Width">
        <Text>{width}</Text>
      </Row>
    </Card>
  )
}
```

Why?

Confusing Classes



Ölbaum
@oscherler

Follow

JavaScript makes me want to flip the table and say “~~Fuck~~ this shit”, but I can never be sure what “this” refers to.

4:03 AM - 30 Oct 2015

1,245 Retweets 1,417 Likes





What?

—
“*Hooks* are a new addition in React 16.8. They let you use state and other React features without writing a class.”





How?



- State Hook

`useState()`

- Effect Hook

`useEffect()`



useState()



WELCOME TO THE COUNTER OF LIFE

1

useState()



```
1 import React from 'react';
2
3 class App extends React.Component {
4   constructor(props) {
5     super(props);
6
7     this.state = {
8       count: 0,
9     };
10  }
11
12  setCount = () => {
13    this.setState(prevState => ({
14      count: prevState.count + 1,
15    }));
16  }
17
18  render() {
19    return (
20      <div>
21        <p>You clicked {this.state.count} times</p>
22        <button onClick={this.setCount}>
23          Click me
24        </button>
25      </div>
26    );
27  }
28 }
29
30 export default App;
```

useState()



```
1 import React from 'react';
2
3 class App extends React.Component {
4   constructor(props) {
5     super(props);
6
7     this.state = {
8       count: 0,
9     };
10  }
11
12  setCount = () => {
13    this.setState(prevState => ({
14      count: prevState.count + 1,
15    }));
16  }
17
18  render() {
19    return (
20      <div>
21        <p>You clicked {this.state.count} times</p>
22        <button onClick={this.setCount}>
23          | Click me
24        </button>
25      </div>
26    );
27  }
28 }
29
30 export default App;
```

```
1 import React, { useState } from 'react';
2
3 function App() {
4   const [count, setCount] = useState(0);
5
6   return (
7     <div>
8       <p>You clicked {count} times</p>
9       <button onClick={() =>
10         | setCount(prevCount => prevCount + 1)}>
11         >
12         | Click me
13       </button>
14     </div>
15   );
16 }
```

useState()



```
7  const [count, setCount] = useState(1)
8  const [dog, setDog] = useState('Corgi');
9  const [book, setBooks] = useState(
10     [
11       {
12         title: 'React Hooks',
13         review: 'Awesome'
14       }
15     ]
16   );
```

useEffect()



```
12  /* ... */
13
14  componentDidMount() {
15    | document.title = `You clicked ${this.state.count} times`;
16  }
17
18  componentDidUpdate() {
19    | document.title = `You clicked ${this.state.count} times`;
20  }
21
22  /* ... */
```

useEffect()



```
12  /* ... */
13
14  componentDidMount() {
15    | document.title = `You clicked ${this.state.count} times`;
16  }
17
18  componentDidUpdate() {
19    | document.title = `You clicked ${this.state.count} times`;
20  }
21
22  /* ... */
```

```
7    /* ... */
8
9    useEffect(() => {
10   | document.title = `You clicked ${count} times`;
11   }, [count])
12
13   /* ... */
```

useEffect()



```
5  
6   useEffect(() => {  
7     console.log("Subscribe to Something");  
8     return function cleanup() {  
9       console.log("Unsubscribe to Something");  
10    };  
11  });  
12
```

useEffect() cleanup



Hooks the right way



- Only call Hooks at the Top Level, not:
 - inside loops X
 - conditions X
 - nested function X



Hooks the right way

```
1 function Form() {
2   // 1. Use the name state variable
3   const [name, setName] = useState('Mary');
4
5   // 2. Use an effect for persisting the form
6   useEffect(function persistForm() {
7     localStorage.setItem('formData', name);
8   });
9
10  // 3. Use the surname state variable
11  const [surname, setSurname] = useState('Poppins');
12
13  // 4. Use an effect for updating the title
14  useEffect(function updateTitle() {
15    document.title = name + ' ' + surname;
16  });
17
18  // ...
19 }
```



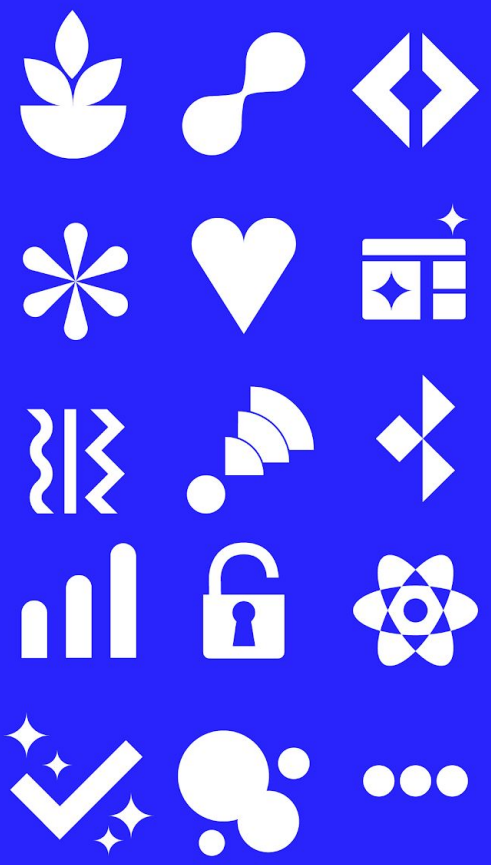

Hooks the right way

- Only call Hooks from React Functions:
 - regular JavaScript functions x
 - custom Hooks v
 - React function components v



References

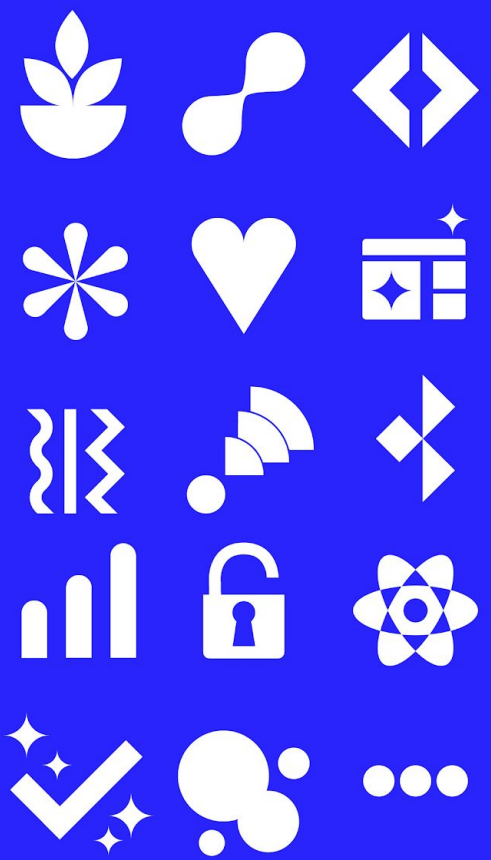
- <https://reactjs.org>
- <https://www.freecodecamp.org/news/learn-the-basics-of-react-hooks-in-10-minutes-b2898287fe5d/>



Q&A

hello@polidea.com

Bē     



Thanks!

hello@polidea.com

Be     