

Topics: React Day 4 (events)

DOM Event and React Syn. Event :

- In a regular DOM event, even after the event propagation, the value will still be located. So we still have a Reference
- But, in the case of Synthetic Event in React,

DOM event:

```
46 const button = document.querySelector("button");
47 let storeEve = null;
48 button.addEventListener("click", function (event) {
49   storeEve = event;
50   console.log("Event happening", storeEve.target);
51   setTimeout(function () {
52     console.log("Event finished", storeEve.target);
53   }, 1000);
54 });
```

React Synthetic event: (on toggle component, refer code)

```
21 let storeEve = null;
22 const handleClick = () => {
23   storeEve = event;
24   console.log("Event happening", storeEve.target);
25   setTimeout(function () {
26     console.log("Event finished", storeEve.target);
27   }, 1000);
28   customOnClick(isSwitchedOn);
29 };
30
31 let innerCircleComponent = null;
32 if (isSwitchedOn === true) {
33   innerCircleComponent = <div style={innerCircleStyle}></div>;
34 }
35 return (
36   <div onClick={handleClick} style={outerCircleStyle}>
37     {innerCircleComponent}
38   </div>
```

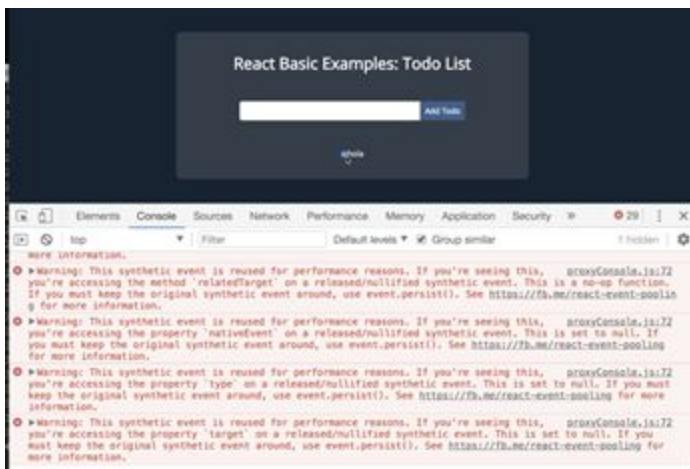
<https://reactjs.org/docs/handling-events.html>

<https://medium.com/@baphemot/understanding-reactjs-setstate-a4640451865b>

<https://stackoverflow.com/questions/49500255/warning-this-synthetic-event-is-re-used-for-performance-reasons-happening-with>

Gives this kind of Error in Synthetic Event:

<https://reactjs.org/docs/events.html>



<https://reactjs.org/docs/handling-events.html>

Synthetic Events in React:

- Used for performance Reason
- Helps in clearing the memory
- These are not stored like we do in Regular DOM.

This keyword Anomaly, State and the Toggle Component:

```
toggle: false // adding in state
};
// const appInstance = this;
// this.handleToggleState = this.handleToggleState.bind(appInstance);
}

handleToggleState = toggleState => { // func to change state
  this.setState({ toggle: !toggleState });
};

render() {
  return (
    <div className="App">
      <Form />
      <Toggle
        isSwitchedOn={this.state.toggle} // passing func as prop
        customOnClick={this.handleToggleState}
      />
    </div>
  );
}
```

<https://medium.com/byte-sized-react/what-is-this-in-react-25c62c31480>

<https://reactjs.org/docs/faq-functions.html>

Steps in Toggle passing fn into the toggle component:

1. Passed the function into the toggle component as a prop
2. Went inside the toggle component
3. Called that func whenever we are having a handleClick event

<https://css-tricks.com/understanding-react-setstate/>


We get an Error because we haven't fixed "this" keyword anomaly!!!

- Normal functions work dynamically, don't care where it is called.
- It would give undefined because we are working with virtual DOM and the context is regular DOM
- SO, we voluntarily need to "bind" it with the Class component (use constructor fn)
- Or, simply use an arrow function. because arrow have lexical scope and don't have their this keyword. In other words, the value of this is determined where it is defined

```
toggle: false
};
// const appInstance = this;
// this.handleToggleState = this.handleToggleState.bind(appInstance);
}

handleToggleState = toggleState => {
  this.setState({ toggle: !toggleState });
};

},
toggle: false
};
const appInstance = this;
this.handleToggleState = this.handleToggleState.bind(appInstance);
}
```



Bind it within Constructor Function

Set Change:

```
// Wrong
this.state.fullName = 'eduardo';

//Correct
this.setState({
  fullName: 'eduardo',
});
```

Last 30 mins → Discussed about the above topics by implementing it to a Form Component

```
1  import React, { Component } from "react";
2
3  class Form extends Component {
4    state = {
5      name: "",
6      email: "",
7      password: ""
8    };
9
10   handleChange = event => {
11     this.setState({ [event.target.name]: event.target.value });
12   };
13
14   handleSubmit = event => {
15     event.preventDefault();
16     console.log(this.state.name, this.state.password, this.state.email);
17   };
18
19   render() {
20     console.log("Rendering");
21     return (
22       <form onSubmit={this.handleSubmit}>
23         <input
24           onChange={this.handleChange}
25           type="text"
26           name="name"
27           value={this.state.name}
28         />
29         <input
30           onChange={this.handleChange}
31           type="email"
32           name="email"
33           value={this.state.email}
34         />
35         <input
36           onChange={this.handleChange}
37           type="password"
38           name="password"
39           value={this.state.password}
40         />
41         <input type="submit" name="" value="Sign up" />
42       </form>
43     );
44   }
45 }
46
47 export default Form;
48
```

Synthetic events are those whose value of event goes away after the event has been fired. True or False?

Attempted
- 38
(70.37%)

EASY



- ☒ True 81.58%
- ☐ False 18.42%

A component's method needs to have its instance value matched by making sure it?

Attempted - 41
(75.93%)

EASY



- ☒ has the this keyword bound 68.29%
- ☐ has the event object bound 31.71%

For input elements, what attribute is essential to make sure the dynamic change is happening?

Attempted - 41
(75.93%)

EASY



- ☐ placeholder 7.32%
- ☒ name 36.59%
- ☐ value 58.54%

How can a child component be able to call a parent component's method?

Attempted - 42
(77.78%)

EASY



- ☒ With props 42.86%
- ☐ With State 57.14%