ReactJS - Event management

Event management is one of the important features in a web application. It enables the user to interact with the application. React support all events available in a web application. React event handling is very similar to DOM events with little changes. Let us learn how to handle events in a React application in this chapter.

Let us see the step-by-step process of handling an event in a React component.

- Define an event handler method to handle the given event.

```
log() {
   cosole.log("Event is fired");
}
```

React provides an alternative syntax using lambda function to define event handler. The lambda syntax is −

```
log = () => {
   cosole.log("Event is fired");
}
```

If you want to know the target of the event, then add an argument **e** in the handler method. React will send the event target details to the handler method.

```
log(e) {
   cosole.log("Event is fired");
   console.log(e.target);
}
```

The alternative lambda syntax is −

```
log = (e) => {
   cosole.log("Event is fired");
   console.log(e.target);
}
```

If you want to send extra details during an event, then add the extra details as initial argument and then add argument **(e)** for event target.

```
log(extra, e) {
   cosole.log("Event is fired");
   console.log(e.target);
   console.log(extra);
   console.log(this);
}
```

The alternative lambda syntax is as follows −

```
log = (extra, e) => {
   cosole.log("Event is fired");
   console.log(e.target);
   console.log(extra);
   console.log(this);
}
```

Bind the event handler method in the constructor of the component. This will ensure the availability of *this* in the event handler method.

```
constructor(props) {
   super(props);
   this.logContent = this.logContent.bind(this);
}
```

If the event handler is defined in alternate lambda syntax, then the binding is not needed. *this* keyword will be automatically bound to the event handler method.

Set the event handler method for the specific event as specified below −

```
<div onClick={this.log}> ... </div>
```

To set extra arguments, bind the event handler method and then pass the extra information as second argument.

```
<div onClick={this.log.bind(this, extra)}> ... </div>
```

The alternate lambda syntax is as follows −

```
<div onClick={this.log(extra, e)}> ... </div>
```