The Error Handling Lifecycle Methods

In this lesson, we'll define the ErrorBoundary component which detects whether any error occurred because of descendant components.

Sometimes things go bad and errors are thrown. The error lifecycle methods are invoked when an error is thrown by a descendant component.

The `ErrorBoundary` Component

Let's implement a simple component to catch errors in the demo app. We'll call this `ErrorBoundary`.

Here's the most basic implementation:

```
import React, { Component } from 'react';
class ErrorBoundary extends Component {
  state = {};
  render() {
    return null;
  }
}
export default ErrorBoundary;
```

Now, let's incorporate the error lifecycle method into our application.

> The given code will not display anything as `render` method returns `null`. However, the component `ErrorBoundary` is added.

```
import React, { Component } from 'react';
class ErrorBoundary extends Component {
  state = {};
  render() {
```

```
    return null;
  }


}
export default ErrorBoundary;
```

In the next lesson, we'll define the `getDerivedStateFromError` for the `ErrorBoundary` component.