Handling Errors

Let's learn how to handle errors when something goes off.

WE'LL COVER THE FOLLOWING

In the real world, things often break, right?

It’s possible that in the process of fetching the lazy-loaded resource, a network error occurs.

Error Boundary

To handle such a case, be sure to wrap your lazy-loaded components in an *error boundary*.

Remember error boundaries from the Lifecycle methods chapter?

Here’s an example:

```
import { Suspense } from 'react'
import MyErrorBoundary from './MyErrorBoundary'
const Scene = React.lazy(() => import('./Scene'))
class Game extends Component {
  state = {
    startGame: false
  }

  render () {
    return <MyErrorBoundary>
         {!this.state.startGame ?
           <GameInstructions /> :
                  <Suspense fallback="loading ...">
                    <Scene />
                  </Suspense>}
                </MyErrorBoundary>
  }
}
export default Game;
```

Now, if an error occurs while fetching the lazy-loaded resource, it'll be graciously handled by the error boundary.

In the next lesson, we'll discuss what to do when named exports are not allowed.