

```
18 |     </li>
19 |   );
20 | };
21 |
22 | WrappedSingleListItem.propTypes = {
23 |   index: PropTypes.number,
24 |   isSelected: PropTypes.bool,
25 |   onClickHandler: PropTypes.func.isRequired,
26 |   text: PropTypes.string.isRequired,
27 | };
28 |
29 | const SingleListItem = memo(WrappedSingleListItem);
30 |
31 | // List Component
32 | const WrappedListComponent = ({
33 |   items,
34 | }) => {
35 |   const [setSelectedIndex, selectedIndex] = useState(); // selectedIndex should be first//
36 |
37 |   useEffect(() => {
38 |     setSelectedIndex(null);
39 |   }, [items]);
40 |
41 |   const handleClick = index => {
42 |     setSelectedIndex(index);
43 |   };
44 |
45 |   return (
46 |     <ul style={{ textAlign: 'left' }}>
47 |       {items.map((item, index) => (
48 |         <SingleListItem
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

+ v ^ x

src > JS index.js > [0] WrappedSingleListItem

```
1
2 import React, { useState, useEffect, memo } from 'react';
3 import PropTypes from 'prop-types';
4
5 // Single List Item
6 const WrappedSingleListItem = ({
7   index,
8   isSelected,
9   onClickHandler,
10  text,
11 }) => {
12   return (
13     <li
14       style={{ backgroundColor: isSelected ? 'green' : 'red' }}
15       onClick={onClickHandler(index)} // it should be onClick={() => onClickHandler(Boolean(index))}
16     >
17       {text}
18     </li>
19   );
20 };
21
22 WrappedSingleListItem.propTypes = {
23   index: PropTypes.number,
24   isSelected: PropTypes.bool,
25   onClickHandler: PropTypes.func.isRequired,
26   text: PropTypes.string.isRequired,
27 };
28
29 const SingleListItem = memo(WrappedSingleListItem);
30
31 // List Component
```

src > JS index.js > items

```
39   }, [items]);
40
41   const handleClick = index => {
42     setSelectedIndex(index);
43   };
44
45   return (
46     <ul style={{ textAlign: 'left' }}>
47       {items.map((item, index) => (
48         <SingleListItem
49           onClickHandler={() => handleClick(index)}
50           text={item.text}
51           index={index}
52           isSelected={selectedIndex}
53         />
54       ))}
55     </ul>
56   )
57 };
58
59 wrappedListComponent.propTypes = {
60   items: PropTypes.array(PropTypes.shapeOf({ // It should be PropTypes.arrayOf and PropTypes.shape
61     text: PropTypes.string.isRequired,
62   })),
63 };
64
65 wrappedListComponent.defaultProps = {
66   items: null,
67 };
68
69 const List = memo(wrappedListComponent);
```