Hook(s)

Hooks are the new feature introduced in the React 16.8 version. It allows you to use state and other React features without writing a class. Hooks are the functions which "hook into" React state and lifecycle features from function components. It does not work inside classes.

Hooks are backward-compatible, which means it does not contain any breaking changes. Also, it does not replace your knowledge of React concepts.

When to use a Hooks

If you write a function component, and then you want to add some state to it, previously you do this by converting it to a class. But, now you can do it by using a Hook inside the existing function component.
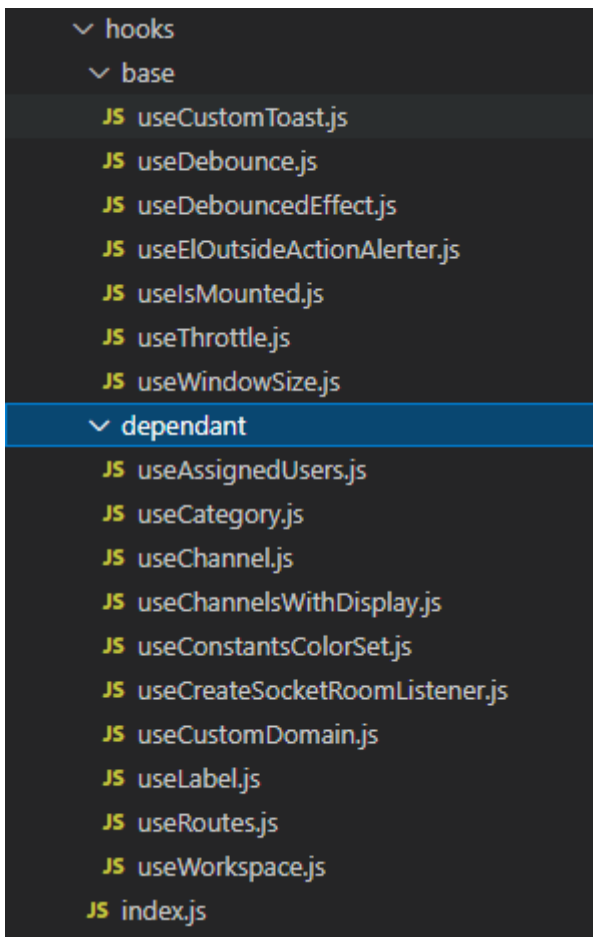
Rules of Hooks

Hooks are similar to JavaScript functions, but you need to follow these two rules when using them. Hooks rule ensures that all the stateful logic in a component is visible in its source code. These rules are:

Folder Structure

- /hooks
  - /category-name (Optional)
    - use{HookName}.js
  - use{HookName}.js
  - index.js

Folder Structure Sample

React Hook File Sample

```javascript
import { useEffect, useState } from "react";

function useWindowSize() {
  ⎕..
}

export default useWindowSize;
```

React Hook Grouping Sample: index.js

```javascript
export { default as useDebouncedEffect } from "@hooks/base/useDebouncedEffect";
export { default as useIsMounted } from "@hooks/base/useIsMounted";
..
```