

useEffect

this arrow function
is callback function.

Syntax-

```
useEffect ( () => { }, [] );
```

Dependency array.

If we need to do something after rendering the component, we have to write inside the `useEffect()`.

Eg. Body.js

```
const Body = () => {
```

```
  useEffect ( () => {
```

```
    console.log ("useEffect Rendered");
```

```
  }, [] );
```

```
  console.log ("Body Rendered");
```

```
  return (
```

```
    ***code**
```

```
  );
```

```
};
```

console:

```
> Body Rendered  
> useEffect Rendered.
```


* API call - fetching Data.

Note - Install - Allow CORS
exten

```
const fetchData = async () => {  
  const data = await fetch("https://...");  
  const json = await data.json();  
};
```

How we use this fetched data, and populate our UI.

⇒ src/components/Body.js

```
const Body = () => {  
  const [listOfRestaurant, setListOfRestaurant] = useState([]);  
  useEffect(() => {  
    fetchData();  
  }, []);  
  const fetchData = async async () => {  
    const data = await fetch("https://...");  
    const json = await data.json();  
    setListOfRestaurant(json?.data?.card[2]?.data?.data  
      ?.card);  
  };  
  return (  
    {JSX code} );  
};
```

optional chaining

* Shimmer UI *

* SignIn-Out *

/src/components/Header.js

```
const Header = () => {  
  const [btn, setBtn] = useState("Login");
```

```
  return (  
    <button  
      onClick={() => {  
        setBtn("log Out");  
      }}  
    >  
      {btn}  
    </button>  
  );  
};
```

```
    <button  
      onClick={() => {  
        btn === "login"  
          ? setBtn("log out")  
          : setBtn("login");  
      }}  
    >  
      {btn}  
    </button>
```


Imp
* Whenever State Variables updated, react triggers a reconciliation cycle (re-renders the component).

* Creating Filter - Search By Name *

./src/component/Body.js

```
const Body = () => {  
  const [resList, setResList] = useState([]);  
  const [searchText, setSearchText] = useState("");
```

```
  { /* Prev. code */ }
```

```
  return (
```

```
    <div className="flex">
```

```
      <input
```

```
        type="text"
```

```
        value={searchText}
```

```
        onChange={ (e) => {
```

```
          setSearchText(e.target.value);
```

```
        } }
```

```
      />
```

```
      <button onClick={ () => {
```

```
        const filteredList = resList.filter((res) =>
```

```
          res.data.name.toLowerCase().includes(  
            searchText.toLowerCase()
```

```
          )
```

```
        );  
        setResList(filteredList); } } > Search
```

```
      </button>
```