

- ☐ import necessary library or self code
 - ☐ create state
 - ☐ useEffect function
 - ☐ api calling function
 - ☐ component code
 - ☐ for css : website:codepen
 - ☐ for toast :npm install react-hot-toast
 - ☐ for loading :loading io
 - ☐
-

full code

1. first execute: `useEffect()` and then data will be saved in `ProductList` then show this data in table.
2. there are a REFRESH button.when this is clicked the `useEffect()` will be called again and fetch.
3. In table there are delete option where user can delete this item.when this button is clicked this call a delete-api and then refresh variable will be increased and again `useEffect()` will be called.

```
import React, { useEffect,useState } from 'react';
import axios from 'axios';
import toast, { Toaster } from 'react-hot-toast';
import Loading from '../loader/Loading'
const readPage = () => {
  const [ProductList,SetProductList] = useState([])

  const [Refresh,SetRefresh] = useState(0);

  const [Load,SetLoad] = useState(true);
  useEffect(()=>{

    GetAPI();

  },[Refresh]);
  const GetAPI = async ()=>{
    let res = await axios.
get("https://crud.teamrabbil.com/api/v1/ReadProduct");
    // console.log(res);
    let product = res.data['data'];
    SetProductList(product);
```



```

        </tr>
      )
    })
  }
  </tbody>
</table>
<Toaster/>
</div>
</div>

</div>
</>
)
}
export default readPage;

```

Import

```

import React, { useEffect,useState } from 'react';
import axios from 'axios';
import toast, { Toaster } from 'react-hot-toast';
import Loading from '../loader/Loading'

```

State

- this is state declaration
 - ProductList is initially empty string
 - Refresh is initially 0 and used for refresh the component
 - Load is initially true and used for loading.
 - this is used in time of calling api.
 - sometimes it takes time to fetch data from database,in this case user see loading animation.

```

const [ProductList,SetProductList] = useState([])
const [Refresh,SetRefresh] = useState(0);
const [Load,SetLoad] = useState(true);

```

useEffect

- ☐ useEffect : in this section (all function) will be executed after the page load means this section will execute first.

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

API calling

- ☐ this is API calling function . here used async await method.
- ☐ this API calling is through axios

```
const API = async () => {  
  let res = await  
  axios.get("https://crud.teamrabbil.com/api/v1/ReadProduct");  
  // console.log(res);  
  let product = res.data['data'];  
  SetProductList(product);  
  // console.log(product);  
}
```

- this is deletion of item
 - take product id as parameter and delete data using axios
[https://crud.teamrabbil.com/api/v1/DeleteProduct/\\${id}](https://crud.teamrabbil.com/api/v1/DeleteProduct/${id})
 - as api is called so there may take time so for loading animation we call a function
SetLoad(true); means loading enable
 - then it check its status
 - it is 'success' then show a success toast msg
toast.success("delete successfully");
 - call SetRefresh(Refresh+1); because data has changed and there need a GetAPI(); calling.that's why we call useEffect() function.
 - it is fail then also show a error toast msg toast.error("delete failed");
 - when it shows error then call: SetLoad(false); because there is no change in data. So no need for waiting.

-

```
const DeleteAPI = async(id)⇒{
  SetLoad(true);
  let res = await
axios.get(`https://crud.teamrabbil.com/api/v1/DeleteProduct/${id}`);
  let deleteItem = res.data['status'];
  if(deleteItem==='success'){
    toast.success("delete successfully");
    SetRefresh(Refresh+1);
  }
  else{
    toast.error("delete failed");
    SetLoad(false);
  }
}
```

main Component

when the Load = true then this will be active:

```
{Load && <Loading />}
```

- when the refresh button is clicked then the refresh value or state will be changed.
- this change effect the useEffect function. because of setting this variable in useEffect such as, `useEffect(()⇒{API();}, [Refresh]);`

```
<button className='btn btn-outline-dark' onClick=
{()⇒SetRefresh(Refresh+1)}>REFRESH</button>
```

- we take delete btn for delete this item from the list.
 - `onClick={()⇒{DeleteAPI(item['_id'])}}`
 - when this btn will be clicked this `DeleteAPI(item['_id'])` this will be called and send item['_id'] as parameter

```
<td><button onClick={()⇒{DeleteAPI(item['_id'])}} className='btn btn-
danger'>Delete</button></td>
```

- we have many data in ProductList. we show this data using loop.
- that's why we use `ProductList.map((item,i)⇒{ return (.....)})`
- `ProductList.length>0 && ProductList.map((item,i)⇒{ return (.....)})` we use this for checking if there are any data or not. if there is no data we don't run loop.
- `<td>{item['_id']}` in this case, we use data from external source.so we take this data by using `{...}` . In Item list we have many keys .that's why we use `item['_id']`
- when Load is true then the Loading component will be shown.
-

```

return (
  <
    {Load && <Loading/>}
    <div className='container'>
      <div className='row'>
        <div className="col-12">
          <h1>Product List</h1>
          <hr />
          <button className='btn btn-outline-dark'
            onClick={()⇒SetRefresh(Refresh+1)}>REFRESH</button>
          <br />
          <div className='table-responsive'>
            <table className="table table-striped">
              <tbody>
                {
                  ProductList.length>0 &&
                  ProductList.map((item,i)⇒{
                    return(
                      <tr>
                        <td>{item['_id']}

```

```
        <Toaster />
      </div>
    </div>
  </div>
</div>
</>
)
```

Loading

```
import React from 'react'
import spinner from '../assets/img/spinner.svg'
const Loading = () => {
  return (
    <div className="container"
      style={{zIndex:100,position:"fixed",marginTop:"100px"}}>
      <div className="row d-flex justify-content-center">
        <div className='col-md-3'>
          <img src={spinner} alt="" />
        </div>
      </div>
    </div>
  )
}
export default Loading;
```