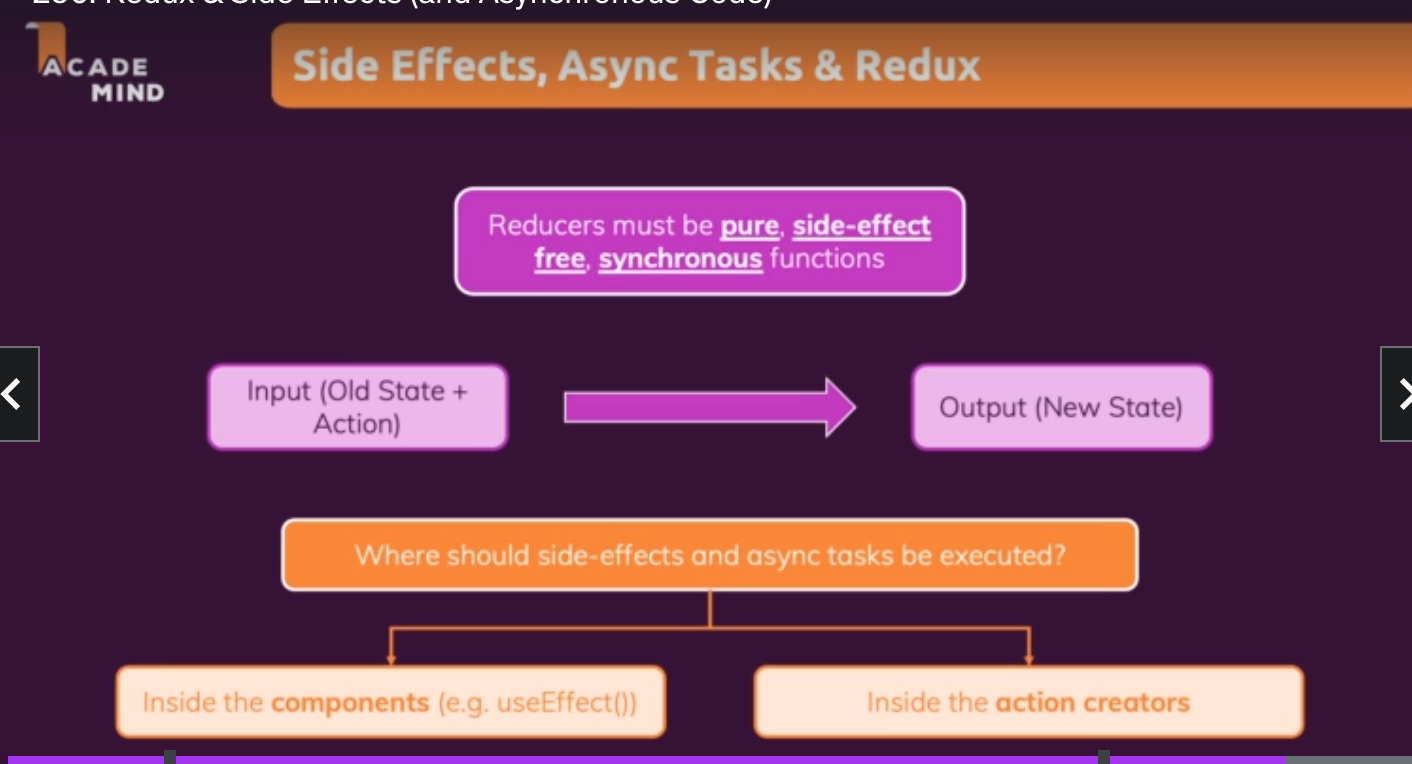
We need to explore following concepts in this notes.

1. Handling Async tasks with redux
2. Where to put our code
3. The redux dev tools.

Redux side effects and asyncronous code.



Since sideEffects like useEffects cannot be used inside reducer functions we need an alternative for us to call async calls.

For more details review this video :- <https://tsys.udemy.com/course/react-the-complete-guide-incl-redux/learn/lecture/25600342#content>.

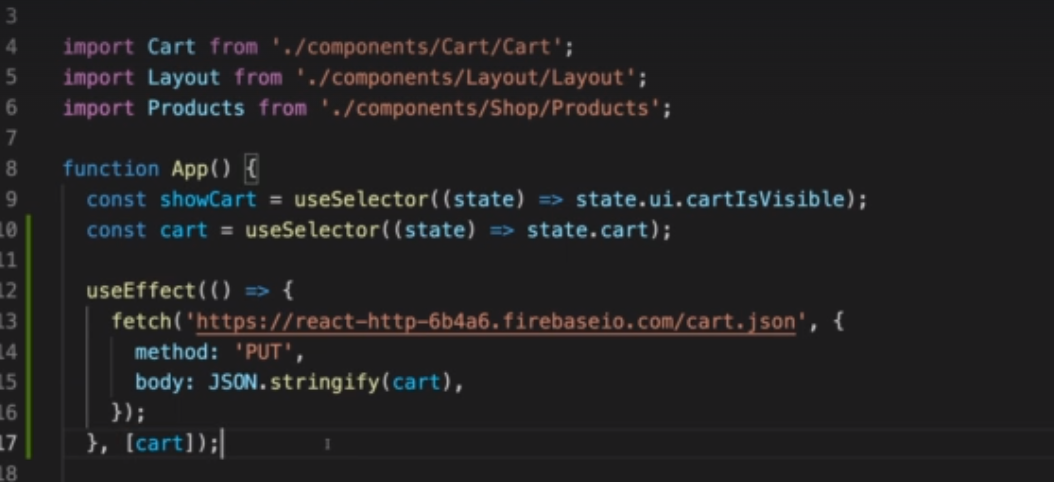
We should never use sideEffects inside a reducer functions.

Hence we can either directly call the async code inside the component or we can call it inside actions creators.

Lets first try with component approach.

Never mutate redux state outside of the reducer function.

Using useEffect() with reducers. We can add a useEffect in the main App.js file



This helps us to keep the reducer logic separate . Our UI will get updated first via state management and since cart updates is a dependency in useEffect, PUT request will be sent to cart.json and database would be updated.

So now with the above code we have kept the data transformation logic in the reducer and the side effect logic inside the component.

There is a problem with the above approach though . Out PUT request would be called everytime our application loads. We also need to handle the http response error using async await which can be done as below.

We can add a new state property called notifications to handle the notifications in uiSlice.



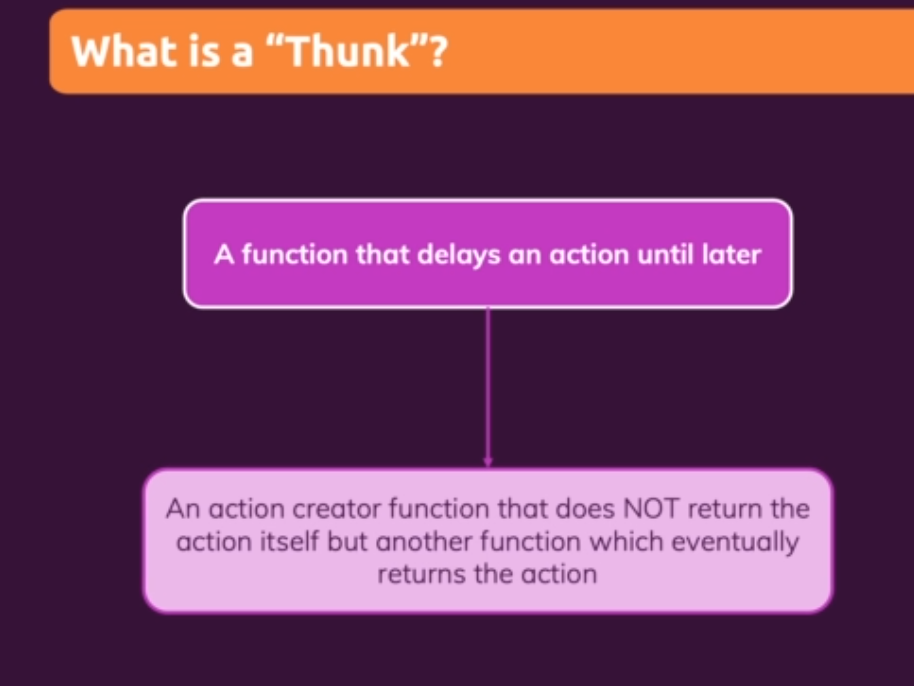
And then we can handle the call in App.js as follows for calling the post reques on App load. IsInitial property is added to avoid the database getting overwritten on application load,





The above approach involved using components and there is another approach to this and that by using redux thunk,(Action creator thunk)

What is a thunk.?



Other approach can be by using reduc thunk for handling Http calls and dispatching the data.

This can help us to keep the components leaner. The entire logic can be outsourced to cartslice.js .









Similarly write the fetch logic refer this video for more reference.

<https://tsys.udemy.com/course/react-the-complete-guide-incl-redux/learn/lecture/25600374#content>

It is important to understand how get and put request along with state changes are done.

Redux devtools:-

Redux devtools have a browser extension. Click on redux and you will have insights to the state of the application.

