Redux Toolkit (RTK)

link for doc (redux-toolkit.js.org)

React (different library)

React-DOM
(different library)
[for browsers]

React-Native
[for mobiles]

April 201

Su	Mo	Tu	We	Th	Fr
		3	4	5	6
8	9	10	11	12	13
15	~~16~~	17	18	19	20
22	23	24	25	26	27
29	30				

It provides the core functionality for managing global state using actions, reducers & the store.

$\boxed{\text{Redux}}$ (Independent state management library)

$\boxed{\text{Purpose}}$: It is a binding library that integrates Redux with react

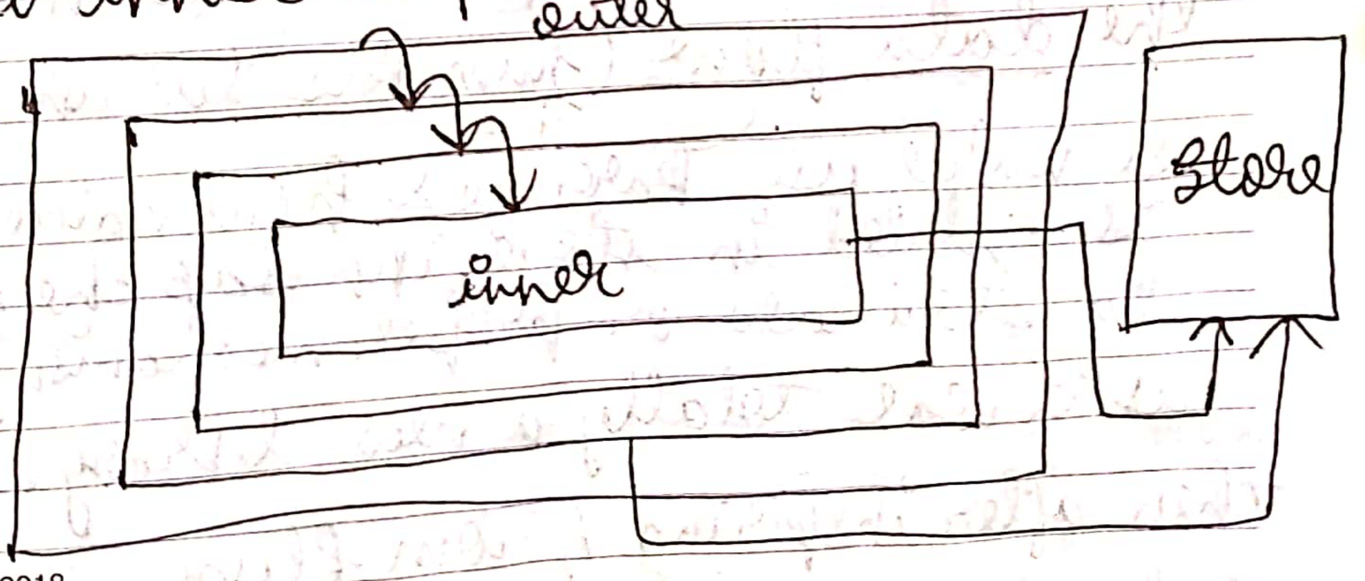
Usage with React → $\boxed{\text{react-redux}}$

It provides hooks (useSelector, useDispatch) and components (<Provider>) to access store & dispatch action

So, if we have to use redux in react, then we have to install redux ~~as well as~~ which is a core-library & the whole binding which have to be used in react known as react-redux.

Why to use Redux?

To pass prop from outer component to inner component.



May

04 FRI

124-241

201

→ we have mini-portions of store too

Store (single source of truth)

reducers
(like controllers where all logics are being written)

useSelector	useDispatch
Hindi:- Jab koi value select karni hai store se	Hindi:- Jab koi value bhejni ho ~~store~~ store ~~mein~~ mein
English:- useSelector can directly communicate with store to bring any info. out of it.	English:- useDispatch is used along with reducer to update info in store

Before redux, flux was created

05 SAT

125-240

as a state-management library but the problem is in the data flow (hum kis tarah se value use karenge & how value is updated in store, inn sab cheezo ka structure or proper mechanism) as it was totally a new library.

06 SUN

Then after inspiring from flux, redux came into picture which was slightly better version of flux.

May 201

Su	Mo	Tu	We	Th	Fr	Sa
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

Terminologies used in Redux

→ Single source of truth (In layman, ek hi store hona chaiye)

In Redux, the single source of truth is a JS object called the store that holds the entire state of an application.
[This means that all data is stored in one place, which makes it easier to manage and debug the application.]

→ Redux state must be read only

In a redux application, we cannot modify application state directly.

→ Changes in redux store must ~~not~~ be made through a function

We cannot directly make changes in store rather changes should be made through pure functions or we call them reducers

So, now there are so many tools like redux-thunk and middlewares which ~~doe~~ helps in debugging.

There is redux dev tools extension now.

But unfortunately, we have to do lot of setup and stuff in redux to do all this

→ more abstract

So Redux-Toolkit was introduced

which was batteries-included toolset [this means it provides simple flow and most of the things it manages internally] for efficient redux development.

Features

- → Built-in middlewares are there (like redux-thunk)
- → Introduced slices so that how to manage reducers

Coding Process

Steps 1) Create app folder inside src folder and create store.js.

Code snippet

↳ destructure configureStore from redux-toolkit

```
import {configureStore} from
        '@reduxjs/toolkit';
export const store = configureStore
        ({ });
```

↓ this configureStore takes object inside it

Step 2) We create reducers in redux-toolkit by creating slices.

So, create new folder inside src known as features.

features can be → login feature
→ product feature
→ todo feature
& many more.... in real projects

We create for todo here, so create todo folder inside features folder as :

src/features/todo

Now, create todoSlice.js file next.

i.e. src/features/todo/todoSlice.js

code snippet

```
import {createSlice, nanoid} from '@reduxjs/toolkit';
```

generates unique ids everytime

```
//We create initialState now.
const initialState = {
        todos: [{id: 1, text: 'Hello World'}]
}
```

Array bhi ho skta hai ya object bhi

```
export const todoSlice = createSlice(
      object ↖ {

              name: 'todo'
              initialState,
it contains properties & function      syntax hai ki
            ↖ reducers: {                ye 2 cheezo
                                        ka access milega
                    addTodo: (state, action) =>
                                                {
                       const todo = {
                         ~~id = 1, text: 'Hello world'~~
                         id: nanoid(),
                         text: action.payload   object
                       }
                       state.todos.push(todo);
                    },
                    removeTodo: (state, action) =>
                                                {
                       state.todos = state.todos
                         filter(todo => todo.id
                                   !== action.
                                        payload)
                                                ;

              }, → we can add more
                   reducers here...
         }
```

June 2018

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Now, we have to export reducers we created above individually so that hume ye components mein kaam aa ske.

i.e. export const {addTodo, removeTodo} = todoSlice.actions

Now, we created store.js in app folder so it also needs awareness ~~that~~ of all reducers ~~/~~ ∵ it is a restrictive store and does not update value by ~~&~~ taking any function.

To jo hum reducers register karenge store mein woh sirf unse value leke hi update karega.

So, we need to export all the reducers ~~in~~

i.e. export default todoSlice.reducer;

Now, In store.js, we ~~use~~ the import this exported reducer.

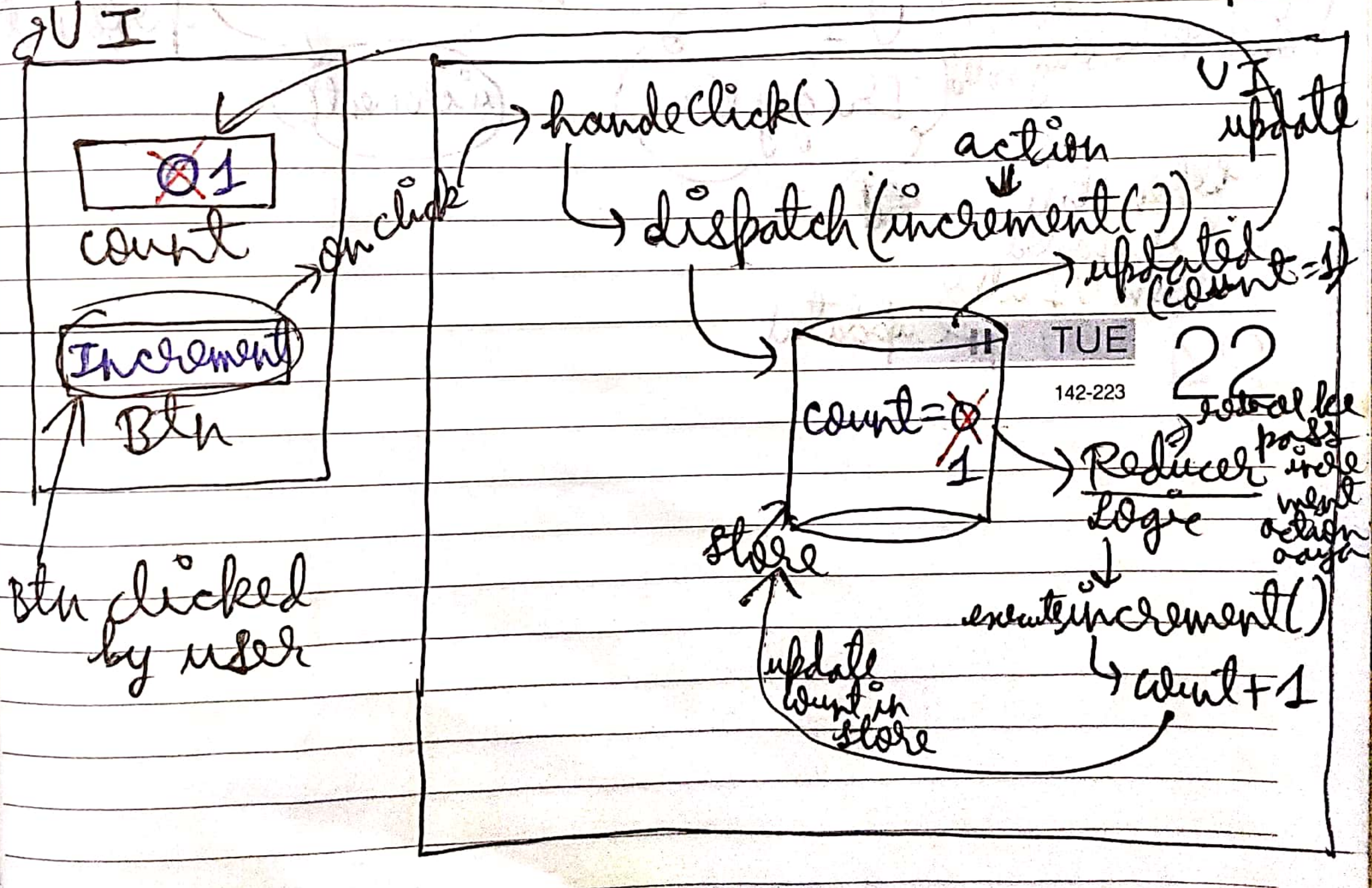
import {configureStore} from '@reduxjs/toolkit';

import todoReducer from '../features/todo/todoSlice';

```
export const store = configureStore({
        reducer: todoReducer
})
```

... Rest I have done in code part

Redux-flow by Love Babbar with count example

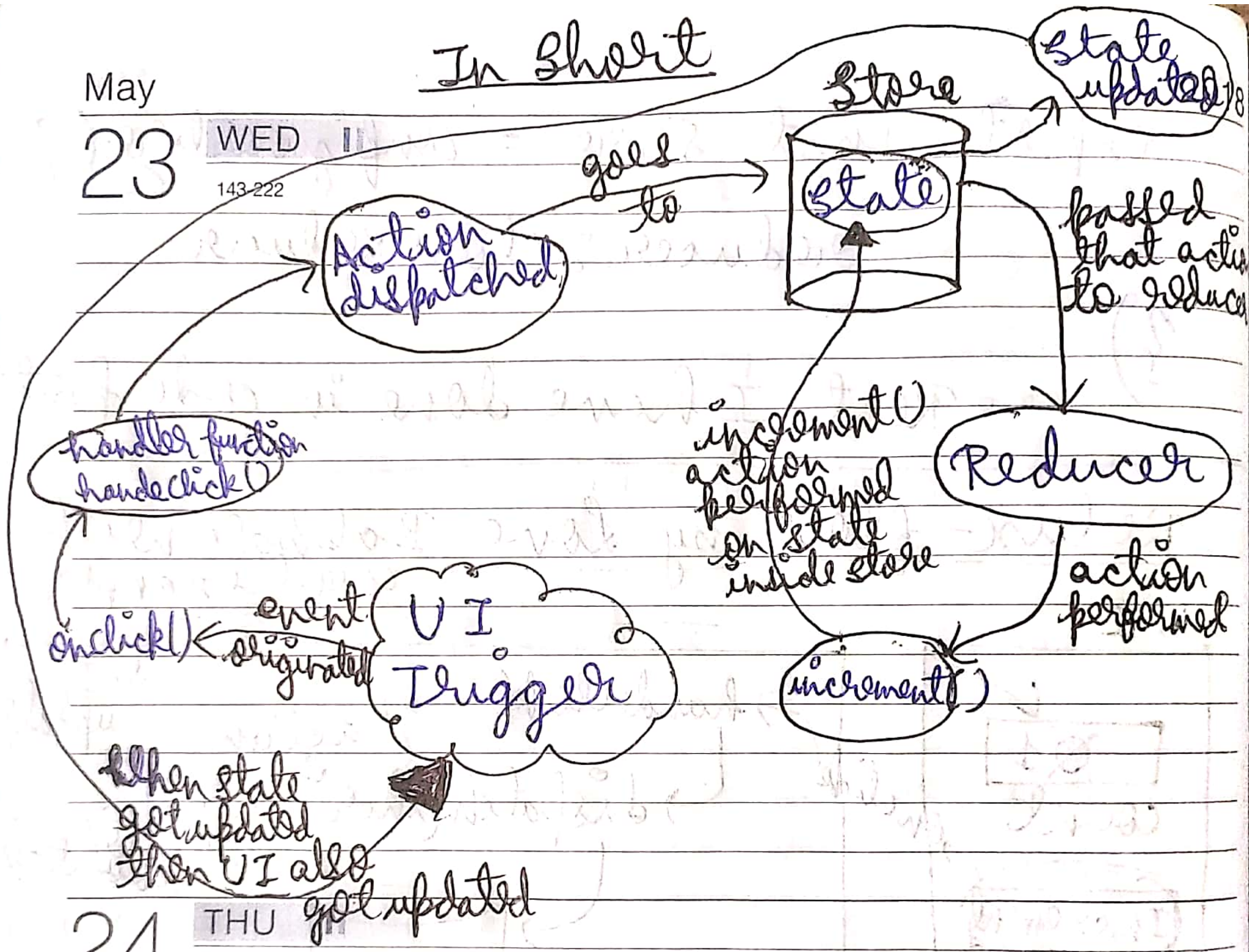


May

23 WED

143-222

In Short



24 THU

144-221