

---

---

---

---

---



- \* npm run dev (to run react using vit)
- \* npm install -D TailwindCSS
  - <sup>↑</sup> it's a dev dependency
  - required for development of the product but not for the final product to work.
- \* import {createSlice} from '@reduxjs/toolkit',  
 import {nanoid} from '@reduxjs/toolkit'  
 to generate unique ids
- + Slice VS Splice which creates mutation?
  - let Array = [1, 2, 3, 4, 5]
  - let Slice = Array.slice(0, 3);
  - let Splice = Array.splice(2, 2, 6, 7)
  - now array = [1, 2, 6, 7, 5]
  - Slice = [1, 2, 3]
  - Splice = [3, 4]
- const initialState = {
 todos: [
 {
 id: nanoid(),
 title: 'Learn React'
 }
 ]
 };
 export const todoSlice = createSlice({
 name: 'todo',
 initialState,
 reducers: {
 addTodo: (state, action) => {
 const todo = {
 id: nanoid(),
 title: action.payload
 };
 state.todos.push(todo);
 }
 }
 });
 // generate unique id
 // Array but Splice modifies the array and changes the state so it mutates

removeTodo : (state, action)  $\Rightarrow$  {  
  state.todos = state.todos.filter ( (todo)  $\Rightarrow$   
    todo.id != action.payload)  
  }  
   $\Rightarrow$  // here for me the list is everything  
  in the list will be there but not  
  the one that matches the id (delete)

export const [addTodo, removeTodo] = todoSlice.actions;  
export default todoSlice.reducer

### store.js

import { configureStore } from '@reduxjs/toolkit'  
import todosReducer from '../features/todoSlice'  
   $\uparrow$  this is actually todosSlice but we just named it  
  something else here.  
export const store = configureStore ({  
  reducer : {  
    todos : todosReducer  
  }  
})

\* under ./assets/components/Todos.js

import React from 'react'  
import { useSelector } from 'react-redux'  
import { useDispatch } from 'react-redux'  
import { removeTodo } from '../features/todoSlice'

function Todos () {  
  const todos = useSelector ( (state)  $\Rightarrow$  state.todos.todos)  
  const dispatch = useDispatch ();  
  return (  

{ todos.map ( (todo)  $\Rightarrow$  (

key = todo.id>{ todo.text }

) ) }  

\Rightarrow dispatch (removeTodo (todo.id)) } } > X </button>

  
  )  
  }  
  }  
  export default Todos