

```
//TODOSLICE
```

```
//app
```

```
import { useDispatch, useSelector } from "react-redux";
```

```
import { addTask } from "../todoSlice";
```

```
import { useState } from "react";
```

```
const App=()=>{
```

```
  const [val,setVal]=useState("");
```

```
  const mytask=useSelector((state)=>state.todos.task);
```

```
  const mydis=useDispatch();
```

```
  const taskAdd=()=>{
```

```
    mydis(addTask({id:Date.now(),task:val}))
```

```
  }
```

```
  let snc=0
```

```
  const ans =mytask.map((key)=>{
```

```
    snc++;
```

```
  return(
```

```
    <>
```

```
    <tr>
```

```
      <td>{snc}</td>
```

```
      <td>{key.task}</td>
```

```
    </tr>
```

```
  </>
```

```
)
```

```
}}
```

```
return(
```

```
  <>
```

```
  <h1>to do app</h1>
```

```
  <input type="text" value={val} onChange={(e)=>{setVal(e.target.value)}}/>
```

```
  <button onClick={taskAdd}>add</button>
```

```
  <hr size="4" color="blue"/>
```

```
  <table>
```

```
    <tr>
```

```
    <th>snc</th>
```

```
    <th>Task</th>
```

```
  </tr>
```

```
    {ans}
```

```
  </table>
```

```
)  
}
```

```
export default App;
```

```
//todoslice
```

```
import { createSlice } from "@reduxjs/toolkit";
```

```
const todoSlice=createSlice({  
  name: "todolist",  
  initialState:{  
    task: []  
  },  
  reducers:{  
    addTask:(state,actions)=>{  
      state.task.push(actions.payload);  
    }  
  }  
})
```

```
export default todoSlice.reducer;  
export const {addTask}=todoSlice.actions;
```

```
//store
```

```
import { configureStore } from "@reduxjs/toolkit";  
import todoReducer from "./todoSlice"
```

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

```
//main
```

```
import React from 'react'
```

```
import ReactDOM from 'react-dom/client'
import App from './App.jsx'
import 'bootstrap/dist/css/bootstrap.min.css';

import { Provider } from 'react-redux';
import store from './store.jsx';

ReactDOM.createRoot(document.getElementById('root')).render(
  <Provider store={store}>
    <App />
  </Provider>
)
```