

RTK step-by-step

1 Create Vite project

>npm create vite@latest

OR

>npx create-vite

ensures Vite 6+, React 19+

<https://legacy.reactjs.org/blog/2020/09/22/introducing-the-new-jsx-transform.html#removing-unused-react-imports>

>npx create-vite

ensures Vite 6+, React 19+

THIS IS BETTER SCAFFOLDING

Cd into project

Npm I

Npm run dev and remove jank

2 Add packages

@reduxjs/toolkit

React-redux

Uuid

Overview on redux and decline in popularity then RTK

3 Create a store and a slice, in JS

CREATE store.js

CREATE componentSlice.js

Create movieSlice then initialState

Export reducer and actions

Import reducer into store

Do not wrap app in provider yet as this will mean more syntax has to be correct for the dev server to run

```
const initialState = {  
  movies: [],  
}  
createSlice({
```

```

    Name:"",
    initialState,
    reducers: {

        addMovie: (state, action)=>{
            state.movies.push(action.payload)
        },
        reducer2: (state, action)=>{

        },

    }
});

```

4 wrap app in provider (main.jsx) and pass the store

Store will auto-import, Provider will not

5 create components, in JSX

List and Input (could be Table and Form)

```

[ already done
  populate initialState in Slice, CREATE
  call uuid for id and key values
]

```

6 useSelector to obtain a slice of state - READ

in List/Table dumb component

Build out at first without remove button - add later

When List component done to read only via useSelector then
add to App.jsx and preview data

7 useDispatch to dispatch an action to the store to update state - UPDATE/DELETE

in Input/Form dumb component

ONLY pass the payload to useDispatch - looks weird, where are the other 2 args? They're inferred.

Payload for add is an object (could write a function constructor)
Payload for delete is the object id,
Add useDispatch to List/Table component, then filter list in slice

8 more items of state? Create more slices