

# ✓ Redux Toolkit + Thunk

## 1. Slice (RTK automatically handles actions + reducers)

```
// features/users/userSlice.js
import { createSlice } from "@reduxjs/toolkit";

const userSlice = createSlice({
  name: "users",
  initialState: {
    loading: false,
    users: [],
    error: null
  },
  reducers: {
    usersLoading: (state) => {
      state.loading = true;
    },
    usersSuccess: (state, action) => {
      state.loading = false;
      state.users = action.payload;
    },
    usersError: (state, action) => {
      state.loading = false;
      state.error = action.payload;
    }
  }
});

export const { usersLoading, usersSuccess, usersError } = userSlice.actions;
export default userSlice.reducer;
```

## 2. Thunk Function (Modern way)

```
// features/users/userThunks.js
import { usersLoading, usersSuccess, usersError } from "./userSlice";

export const fetchUsers = () => async (dispatch) => {
  dispatch(usersLoading());

  try {
    const res = await fetch("https://jsonplaceholder.typicode.com/users");
    const data = await res.json();

    dispatch(usersSuccess(data));
  } catch (err) {
    dispatch(usersError(err.message));
  }
}
```

```
    }
};
```

---

### 3. Configure Store (RTK modern setup)

```
// store.js
import { configureStore } from "@reduxjs/toolkit";
import userReducer from "./features/users/userSlice";

const store = configureStore({
  reducer: {
    users: userReducer
  }
});

export default store;
```

---

### 4. React Component

```
// App.jsx
import React, { useEffect } from "react";
import { useDispatch, useSelector } from "react-redux";
import { fetchUsers } from "./features/users/userThunks";

function App() {
  const dispatch = useDispatch();
  const { loading, users, error } = useSelector((state) => state.users);

  useEffect(() => {
    dispatch(fetchUsers());
  }, []);

  if (loading) return <h3>Loading...</h3>;
  if (error) return <h3>Error: {error}</h3>;

  return (
    <div>
      <h2>User List</h2>
      {users.map((u) => (
        <p key={u.id}>{u.name}</p>
      ))}
    </div>
  );
}

export default App;
```

