

createSlice

createSlice là sự kết hợp của createReducer và createAction

Mình khuyên các bạn nên dùng createSlice thay vì createReducer vì các bạn không cần tạo action, action sẽ tự động generate ra cho các bạn.

```
import { createSlice } from '@reduxjs/toolkit'
import type { PayloadAction } from '@reduxjs/toolkit'

interface CounterState {
  value: number
}

const initialState = { value: 0 } as CounterState

const counterSlice = createSlice({
  name: 'counter', // Đây là prefix cho action type của bạn
  initialState, // Giá trị khởi tạo state cho reducer, cũng có thể là function khởi tạo
  reducers: {
    // key name sẽ được dùng để generate ra action
    increment(state) {
      state.value++
    },
    decrement(state) {
      state.value--
    },
    incrementByAmount(state, action: PayloadAction<number>) {
      state.value += action.payload
    }
  }
})

// export action được generate ra từ slice
export const { increment, decrement, incrementByAmount } = counterSlice.actions

// export reducer được generate ra từ slice
export default counterSlice.reducer
```

Với **reducers** trên thì ta không dùng được với những trường hợp

- default case
- matcher case

=> Nên dùng **extraReducers** trong 2 trường hợp này

Vì đặc tính tự động generate ra action khi dùng **reducers** nên nếu chúng ta sử dụng một thunk thì không nên check trong **reducers** mà hãy check trong **extraReducers**

extraReducers

`extraReducers` cũng giống `reducers` nhưng nó sẽ không generate ra actions. `extraReducers` cho phép dùng một số tính năng như `addMatcher` hay `addDefaultCase`

`extraReducers` chính xác giống như reducer trong `createReducer()`

```
import { createAction, createSlice, Action, AnyAction } from '@reduxjs/toolkit'
const incrementBy = createAction<number>('incrementBy')
const decrement = createAction('decrement')

interface RejectedAction extends Action {
  error: Error
}

function isRejectedAction(action: AnyAction): action is RejectedAction {
  return action.type.endsWith('rejected')
}

createSlice({
  name: 'counter',
  initialState: 0,
  reducers: {},
  extraReducers: (builder) => {
    builder
      .addCase(incrementBy, (state, action) => {})

      .addCase(decrement, (state, action) => {})
      .addMatcher(isRejectedAction, (state, action) => {})
      .addDefaultCase((state, action) => {})
  }
})
```

`extraReducers` cho phép dùng cú pháp "map object" nhưng với typescript thì chúng ta nên dùng builder callback

Tóm lại khi nào dùng reducers, khi nào dùng extraReducers

Dùng reducers khi muốn

- generate ra action

Dùng extraReducers khi

- Không muốn generate action
- muốn dùng `addMatcher`, `addDefaultCase`
- Khi dùng với `createAsyncThunk`