

redux toolkit - Google Search

createReducer | Redux Toolkit

+

redux-toolkit.js.org/api/createReducer

Apps Gmail XL 1 UB_Caregiver_Issue... XL 2 3000 3001 My Drive - Google... YouTube Bitbucket WhatsApp git repos Reading list

Getting StartedTutorialsUsage GuideAPIRTK QueryGitHub

here we export these function
increment,decremnet and
incremenetByAmount

we use these function in our code and
for example we dispatch the
increment like these
dispatch(increment());
then builder.addCase will exguate the
and the state will be update

NOTE : here we manully need to
create the action.type

and One great thing about these is

we can dispatch a function before the
action creater like
dispatch(()=>{
 (dispatch)=>{
 dispatch(increment())
 }
})

NOTE: we no need of installing anythink libarary from outside, redux toolkit itsself support these one

```
import { createAction, createReducer } from '@reduxjs/toolkit'

const increment = createAction('counter/increment')
const decrement = createAction('counter/decrement')
const incrementByAmount = createAction('counter/incrementByAmount')

const initialState = { value: 0 }

const counterReducer = createReducer(initialState, (builder) => {
  builder
    .addCase(increment, (state, action) => {
      state.value++
    })
    .addCase(decrement, (state, action) => {
      state.value--
    })
    .addCase(incrementByAmount, (state, action) => {
      state.value += action.payload
    })
})
```

Overview

Usage with the "Builder
Callback" Notation

Parameters

Example Usage

Builder Methods

builder.addCase

builder.addMatche

r

builder.addDefault

Case

Usage with the "Map
Object" Notation

Parameters

Returns

Example Usage

Activate Windows

Go to Settings to activate Windows.

Matchers and Default

07:14 AM
15-02-2022

redux toolkit - Google Search

createReducer | Redux Toolkit

+

redux-toolkit.js.org/api/createReducer

AppsGmailXL 1UB_Caregiver_Issue...XL 230003001My Drive - Google...YouTubeBitbucketWhatsAppgit reposReading list

Redux Toolkit

Getting StartedTutorialsUsage GuideAPIRTK QueryGitHub

Search

configureStore

getDefaultMiddleware

Immutability Middleware

Serializability Middleware

Reducers and Actions

createReducer

createAction

createSlice

createAsyncThunk

createEntityAdapter

Other

createSelector

Matching Utilities

Other Exports

RTK Query

RTK Query Overview

Comparison with Other Tools

Usage With TypeScript

Using RTK Query

API Reference

```
import { createAction, createReducer } from '@reduxjs/toolkit'

const increment = createAction('increment')
const decrement = createAction('decrement')

function isActionWithNumberPayload(action) {
  return typeof action.payload === 'number'
}

const reducer = createReducer(
  {
    counter: 0,
    sumOfNumberPayloads: 0,
    unhandledActions: 0,
  },
  (builder) => {
    builder
      .addCase(increment, (state, action) => {
        // action is inferred correctly here
        state.counter += action.payload
      })
      // You can chain calls, or have separate `builder.addCase()` lines each ti
      .addCase(decrement, (state, action) => {
        state.counter -= action.payload
      })
      // You can apply a "matcher function" to incoming actions
      .addMatcher(isActionWithNumberPayload, (state, action) => {})
      // and provide a default case if no other handlers matched
      .addDefaultCase((state, action) => {})
  })
)
```

Priority

1st addcase() ravi

then

2nd .addmatcher() ravi

then

last lo addDefaultCase()

ravi

mention in documtation

one of the best Advance of Builder notation is we can match the action type dynamically (see the eupocming photos u will understand what i mean)

internally builder is using the switch statemnet

There .addMatcher function is used to match the dynamically the action type and state will be update

while .addDeaultCase() will exguate when no case is matched

NOTE : to update or call the Builder function we need to dispatch the incremenet and decrement function then automaticall the buildeer function will be called and suitable addcase() willl be matched and state willl be updated

redux toolkit - Google Search

createReducer | Redux Toolkit

Demo React

redux-toolkit.js.org/api/createReducer

AppsGmailXL 1UB_Caregiver_Issue...XL 230003001My Drive - Google...YouTubeBitbucketWhatsAppgit reposReading list

Redux Toolkit

Getting StartedTutorialsUsage GuideAPIRTK QueryGitHub

configureStore

getDefaultMiddleware

Immutability Middleware

Serializability Middleware

Reducers and Actions

createReducer

createAction

createSlice

createAsyncThunk

createEntityAdapter

Other

createSelector

Matching Utilities

Other Exports

RTK Query

RTK Query Overview

Comparison with Other Tools

Usage With TypeScript

Using RTK Query

API Reference

```
import { createAction, createReducer } from '@reduxjs/toolkit'

const initialState = {}
const resetAction = createAction('reset-tracked-loading-state')

function isPendingAction(action) {
  return action.type.endsWith('/pending')
}

const reducer = createReducer(initialState, (builder) => {
  builder
    .addCase(resetAction, () => initialState)
    // matcher can be defined outside as a type predicate function
    .addMatcher(isPendingAction, (state, action) => {
      state[action.meta.requestId] = 'pending'
    })
    .addMatcher(
      // matcher can be defined inline as a type predicate function
      (action) => action.type.endsWith('/rejected'),
      (state, action) => {
        state[action.meta.requestId] = 'rejected'
      }
    )
    // matcher can just return boolean and the matcher can receive a generic argument
    .addMatcher(
      (action) => action.type.endsWith('/fulfilled'),
      (state, action) => {
        state[action.meta.requestId] = 'fulfilled'
      }
    )
})
```

here happening is when we

```
dispatch({
  type: "harish/rejected",
  payload: {
    name: "harish",
    year: "2022"
  }
})
```

in our code then store will be called i.e builder() will be exguate the store will check the action type and here we passed the type ending with "/rejected" then these add case is matched and these state will be update

NOTE; the theme is, here we are dynamically matching the action type and state is updating

redux toolkit - Google Search

createSlice | Redux Toolkit

Demo React

redux-toolkit.js.org/api/createSlice

AppsGmailXL 1UB_Caregiver_Issue...XL 230003001My Drive - Google...YouTubeBitbucketWhatsAppgit reposReading list

Redux Toolkit

Getting StartedTutorialsUsage GuideAPIRTK QueryGitHub

configureStore
getDefaultMiddleware
Immutability Middleware
Serializability Middleware
Reducers and Actions
createReducer
createAction
createSlice
createAsyncThunk
createEntityAdapter
Other
createSelector
Matching Utilities
Other Exports
RTK Query
RTK Query Overview
Comparison with Other Tools
Usage With TypeScript
Using RTK Query
API Reference

```
import { createAction, createSlice } from '@reduxjs/toolkit'

const incrementBy = createAction('incrementBy')
const decrement = createAction('decrement')

function isRejectedAction(action) {
  return action.type.endsWith('rejected')
}

createSlice({
  name: 'counter',
  initialState: 0,
  reducers: {},
  extraReducers: (builder) => {
    builder
      .addCase(incrementBy, (state, action) => {
        // action is inferred correctly here if using TS
      })
      // You can chain calls, or have separate `builder.addCase()` lines each
      .addCase(decrement, (state, action) => {})
      // You can match a range of action types
      .addMatcher(
        isRejectedAction,
        // `action` will be inferred as a RejectedAction due to isRejectedAction
        (state, action) => {}
      )
      // and provide a default case if no other handlers matched
      .addDefaultCase((state, action) => {})
  },
})
```

Parameters

when we use createSlice() we need to use builder callback function in the extraReducers place lo ravalu

here addCase() and .addMatcher() and .addDefaultcase() function will exguate same as it i explained the the previous photo , just here we are using in the extraReduers lo, anthi, no new thing here

here action type is matched dynmaically and store is updated

We recommend using this API as it has better TypeScript support (and thus, IDE autocomplete even for JavaScript users), as it

configureStore
getDefaultMiddleware
Immutability Middleware
Serializability Middleware

Reducers and Actions

createReducer
createAction
createSlice
createAsyncThunk
createEntityAdapter

Other

createSelector
Matching Utilities
Other Exports

RTK Query

RTK Query Overview
Comparison with Other Tools
Usage With TypeScript

Using RTK Query

API Reference

```
import { createSlice, createAction } from '@reduxjs/toolkit'
import { createStore, combineReducers } from 'redux'

const incrementBy = createAction('incrementBy')
const decrementBy = createAction('decrementBy')

const counter = createSlice({
  name: 'counter',
  initialState: 0,
  reducers: {
    increment: (state) => state + 1,
    decrement: (state) => state - 1,
    multiply: {
      reducer: (state, action) => state * action.payload,
      prepare: (value) => ({ payload: value || 2 }), // fallback if the payload is not provided
    },
  },
  // "builder callback API", recommended for TypeScript users
  extraReducers: (builder) => {
    builder.addCase(incrementBy, (state, action) => {
      return state + action.payload
    })
    builder.addCase(decrementBy, (state, action) => {
      return state - action.payload
    })
  },
})

const user = createSlice({
  name: 'user',
  initialState: { name: '', age: 20 },
  reducers: {
```

Parameters

initialState

its mandantory to use the
BuilderCallback function in the
extraReducers lo place lo ni

ier callback"

object"

redux toolkit - Google Search

createAsyncThunk | Redux Toolkit

Demo React

how to scroll using keyboard - G

+

← → ↺

redux-toolkit.js.org/api/createAsyncThunk

🔍 🔖 ⚙️ H ⋮

📱 Apps

📧 Gmail

📄 XL 1

📄 UB_Caregiver_Issue...

📄 XL 2

📄 3000

📄 3001

📄 My Drive - Google...

📺 YouTube

📁 Bitbucket

📞 WhatsApp

📁 git repos

»

📖 Reading list

Redux Toolkit

Getting Started Tutorials Usage Guide API RTK Query GitHub

🔍

configureStore

getDefaultMiddleware

Immutability Middleware

Serializability Middleware

Reducers and Actions

createReducer

createAction

createSlice

createAsyncThunk

createEntityAdapter

Other

createSelector

Matching Utilities

Other Exports

RTK Query

RTK Query Overview

Comparison with Other Tools

Usage With TypeScript

Using RTK Query

API Reference

```
import { createAsyncThunk, createSlice } from '@reduxjs/toolkit'
import { userAPI } from './userAPI'

// First, create the thunk
const fetchUserId = createAsyncThunk(
  'users/fetchByIdStatus',
  async (userId, thunkAPI) => {
    const response = await userAPI.fetchById(userId)
    return response.data
  }
)

// Then, handle actions in your reducers:
const usersSlice = createSlice({
  name: 'users',
  initialState: { entities: [], loading: 'idle' },
  reducers: {
    // standard reducer logic, with auto-generated action types per reducer
  },
  extraReducers: (builder) => {
    // Add reducers for additional action types here, and handle loading state a
    builder.addCase(fetchUserId.fulfilled, (state, action) => {
      // Add user to the state array
      state.entities.push(action.payload)
    })
  },
})

// Later, dispatch the thunk as needed in the app
dispatch(fetchUserId(123))
```

Now we are talking about creaAsyncThunk(API)

There is nothing new to say about createAsyncThunk , becoz its work same as it as we are using in our project but now we are using createAsyncThunk with buillder callback notation now, but createAsyncThunk will work same as it as here we are using in our project

here we can added .pending and .rejected to addCase and when these matchs, then callback fnction willl be exguated (Note here .pending and .rejected are using when api repose is rejected and pendded from server)

NOTE: in builder callback notation we can use all the property which i told like ..addMatcher and .addDefaultCase , these work same as i mention and explained in the previos photo