

createAction

`createAction` là một helper function dùng để tạo một Redux action

```
function createAction(type, prepareAction?)
```

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

const increment = createAction<number | undefined>('counter/increment')

let action = increment()
// { type: 'counter/increment' }

action = increment(3)
// returns { type: 'counter/increment', payload: 3 }

console.log(increment.toString())
// 'counter/increment'

console.log(`The action type is: ${increment}`)
// 'The action type is: counter/increment'
```

Sử dụng Prepare callback để tính chỉnh payload

Mặc định bạn truyền vào gì thì payload sẽ là cái đó, trong trường hợp bạn muốn truyền vào x nhưng payload là x + 2 thì bạn có thể dùng prepare function callback

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

const addTodo = createAction('todos/add', function prepare(text: string) {
  return {
    payload: {
      text,
      id: nanoid(),
      createdAt: new Date().toISOString()
    }
  }
})

console.log(addTodo('Write more docs'))
/**
 * {
 *   type: 'todos/add',
 *   payload: {
 *     text: 'Write more docs',
```

```
*      id: '4AJvwMSWEHCchcWYga3dj',
*      createdAt: '2019-10-03T07:53:36.581Z'
*    }
*  }
**/
```

Sử dụng với createReducer()

Bởi vì action creator được return từ `createAction()` có method `toString()` bị override rồi, nên ta có thể dễ dàng dùng với `createReducer()`

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

const increment = createAction<number>('counter/increment')
const decrement = createAction<number>('counter/decrement')

const counterReducer = createReducer(0, (builder) => {
  builder.addCase(increment, (state, action) => state + action.payload)
  builder.addCase(decrement, (state, action) => state - action.payload)
})
```