

Reducer在redux中的作用为，接受dataStore中此时的状态，以及一个action，返回一个新的状态对象进而更新Redux的dataStore.

在此基础上，Reducer Enhancer是定义的一种方法，这个方法接收reducer作为参数，对其进行加工，并且返回一个新的reduer。由于reducer的本质是接受两个参数(storeData, action)的方法，所以经过Reducer Enhancer加工以后返回的是一个reducer,那么也代表Reducer Enhancer返回的是一个接受两个参数(storeData, action)的方法. 它接受代表现有状态的storeData以及Action，返回代表新状态的对象。

例如：

export function customReducerEnhancer(originalReducer){//此处customReducerEnhancer为ReducerEnhancer。它为一个方法，接受需要加工的reducer作为参数

let initialState = null;

return (storeData, action)=> { //ReducerEnhacner最终返回一个新的reducer,也就是一个接受两个参数(storeData, action)的方法，按照reducer性质，其返回现storeData经过action过后新的状态对象。

if (action.type === STORE\_RESET && initialData != null){

return initialState

}

else{

const result = originalReducer(storeData, action);

if(initialState === null){

initialState = result

}

return result

}

}

}

Reducer Enhancer的作用是action在达到reducer前先经过Reducer Enhancer的预处理。可以达到在Reducer Enhancer中对action进行先行拒绝，添加属性等，再分派到reducer中。

Redux的combineReducers便是Redux自带的Reducer Enhancer, 它的作用是将action分发给多个Reducer处理。 Reducer Enhancer对于Redux的dataStore来说是透明的，dataStore并不会知道是否采用了Reducer Enhancer. 它总体遵循的流程还是，action最终交付到Reducer，然后进而reducer更新dataStore.