有一些编写actions的好习惯：

1 - 前期——在开始开发功能之前编写编写action，以便理解功能和知识点

2 - 分类——基于事件资源对actions进行分类

3 - 编写更多——action的编写容易，所以你可以编写更多的actions,来更好的表达应用流程

4 - 事件-驱动——捕获事件而不是命令，因为你要分离事件的描述和事件的处理

5 - 描述——提供针对唯一事件的上下文，其中包含可用于帮助开发人员进行调试的更详细信息

一、操作类型枚举

export enum AuthActionsTypes {

Login = '[Auth] Login',

Logout = '[Auth] Logout',

}

二、定义操作类

export class Login implements Action {

public readonly type = AuthActionsTypes.Login;

}

export class Logout implements Action {

public readonly type = AuthActionsTypes.Logout;

}

三、可辨识联合（Discriminated Unions）

export type AuthActionsUnion = Login | Logout

1- type 可以声明基本类型别名，联合类型，元组等类型

// 基本类型别名

type Name = string

// 联合类型

interface Dog { wong(); }

interface Cat { miao(); }

type Pet = Dog | Cat

// 具体定义数组每个位置的类型

type PetList = [Dog, Pet]

2 - 联合类型用来声明那些可以存储多种类型值的变量

联合类型（Union Types）可以通过管道(|)将变量设置多种类型，

赋值时可以根据设置的类型来赋值。

例： var val : string|number

3 - 可辨识联合（Discriminated Unions）

你可以合并单例类型，联合类型，类型保护和类型别名来创建一个叫做 可辨识联合的高级模式，

它也称做 标签联合或 代数数据类型。 可辨识联合在函数式编程很有用处。

一些语言会自动地为你辨识联合；而TypeScript则基于已有的JavaScript模式。 它具有3个要素：

(1) - 具有普通的单例类型属性— 可辨识的特征。

(2) - 一个类型别名包含了那些类型的联合— 联合。

(3) - 此属性上的类型保护。

首先我们声明了将要联合的接口Login,Logout。 每个接口都有 type属性但有不同的字符串字面量类型。

type属性称做 可辨识的特征或 标签。 其它的属性则特定于各个接口。

注意，目前Login,Logout接口间是没有联系的。 下面我们把它们联合到一起：

export type AuthActionsUnion = Login | Logout

4 - 现在我们使用可辨识联合:

export function authReducer( state: AuthState = initialState, action: AuthActionsUnion): AuthState {

switch (action.type) {

case AuthActionsTypes.SetCurrentUser: {

return merge(state, { currentUser: action.payload.user });

}

default: {

return state;

}

}

}

四、定义操作类的对象常量

（1） - 箭头函数

export const login = (): Login => new Login();

相当于

export const login = function(): Login {

return new Login();

}

export const logout = (): Logout => new Logout();

五、定义操作类的创建者对象

export const authActionCreators = {

login,

logout

};

1 - authActionCreators是一个对象字面量

es6中如果想使用表达式外面的变量名作为成员名，变量的值作为成员值，可进一步简写为

当属性名称和变量名称相同的时候可以只写一个,成为增强对象字面量

var dynamicVar="dyna";

var person={

dynamicVar, //这是一个语法糖，js引擎会解释为dynamicVar:'dyna'

age:15

}

上面

export const authActionCreators = {

login：login ,

logout：logout

};

使用方法

this.store.dispatch(authActionCreators.login());