一、功能

**1、 feature state**

export interface Title { code?: string; name?: string;}

export interface TitlesState extends EntityState<Title> {}

**2、reducer**

export const titlesAdapter = createEntityAdapter<Title>({

selectId: (title: Title) => title.code,

sortComparer: false

});

export const initialState: TitlesState = titlesAdapter.getInitialState({});

const titleReducer = createReducer<TitlesState>(

initialState,

on(loadTitlesSuccess, (state, {titles}) =>

titlesAdapter.addAll(titles, state)

),

);

export function reducer(state: TitlesState, action: Action) {

return titleReducer(state, action);

}

**二 - 功能模块**

**1 -**

E:\计算机\spartacus\projects\core\src\user\store\user-store.module.ts

import { metaReducers, reducerToken, reducerProvider } from './reducers/index';

import { StateModule } from '../../state/state.module';

@NgModule({

imports: [

CommonModule,

ReactiveFormsModule,

StateModule,

StoreModule.forFeature(USER\_FEATURE, reducerToken, { metaReducers }),

EffectsModule.forFeature(effects),

RouterModule,

],

providers: [reducerProvider],

})

export class UserStoreModule {}

**2 - State**

export const USER\_FEATURE = 'user';

export interface StateWithUser { [USER\_FEATURE]: UserState;}

export interface UserState {

account: UserDetailsState;

addresses: LoaderState<Address[]>;

titles: TitlesState;

}

**3 - reducer**

export const reducerToken: InjectionToken<ActionReducerMap<UserState>> = new InjectionToken<ActionReducerMap<UserState>>('UserReducers');

export const reducerProvider: Provider = { provide: reducerToken, useFactory: getReducers,};

export function clearUserState( reducer: ActionReducer<any>): ActionReducer<any> {

return function (state, action) {

if (action.type === AuthActions.LOGOUT) {

state = undefined;

}

return reducer(state, action);

};

}

export const metaReducers: MetaReducer<any>[] = [clearUserState];

export function getReducers(): ActionReducerMap<UserState> {

return {

account: combineReducers({

details: fromUserDetailsReducer.reducer,

}),

addresses: loaderReducer<Address[]>(

USER\_ADDRESSES,

fromAddressesReducer.reducer

),

titles: fromTitlesReducer.reducer,

resetPassword: fromResetPasswordReducer.reducer,

};

}