**Step 1 >>**

npm install @ngrx/store –save

add in app module import file

StoreModule.forRoot(reducerName, {metaReducerName}),

**Step 2 >>**

npm install @ngrx/effects –save

add in app module import file

EffectsModule.forRoot(EffectName),

**Step 3 >>**

npm install @ngrx/store-devtools –save

add in app module import file

StoreDevtoolsModule.instrument({

      maxAge: 25,

      logOnly: environment.production

    })

npm i ngrx-store-localstorage

**Step 4 >>**

Create first state folder in which part module, you want to create and show, like auth

In this folder we have to create action, effects, reducers, types and other folder with their respective files.

**Core/actions/auth.actions.ts**

import { createAction, props } from "@ngrx/store";

import { LoginTypes } from "../config/auth.types";

export const getLoginAction = createAction(

    '[Auth] get login details',

    props<{payload: LoginTypes}>()

);

export const saveLoginAction = createAction(

    '[Auth] save login details',

    props<{payload: any}>()

);

**Core/config/auth.types.ts**

export interface AuthType {

    roleDetails: AuthRoleType,

    isAuth: boolean,

    userName: string,

    accessPagesList: AuthAccessPagesType[]

}

export interface LoginType {

    userName: string,

    password: string,

}

export interface AuthRoleType {

    role?: string,

    department?: string,

}

export interface AuthAccessPagesType {

    pageId?: string,

    pageName?: string,

}

**Core/config/auth.effects.ts**

import { sendEmptyAction } from './../../../core/state/actions/common.actions';

import { AuthService } from './../services/auth.service';

import { Injectable } from "@angular/core";

import { Actions, createEffect, ofType } from "@ngrx/effects";

import \* as authAction from "../actions/auth.actions";

import {map, switchMap} from "rxjs/operators"

import { Store } from '@ngrx/store';

@Injectable()

export class AuthEffects {

    constructor(private \_actions$: Actions, private \_authService:AuthService, private \_store:Store<any>){}

    getLogin$ = createEffect(()=>

        this.\_actions$.pipe(

            ofType(authAction.getLoginAction.type),

            switchMap(({payload})=>{

                return this.\_authService.login(payload).pipe(

                    map((data:any)=>{

                        let json = JSON.parse(JSON.stringify(data));

                        if(json == undefined){

                            // this.\_store.dispatch(

                            //     authAction.saveLoginAction({payload: json})

                            // )

                            return sendEmptyAction();

                        } else {

                            return authAction.saveLoginAction({payload: json});

                        }

                    })

                )

            })

        )

    )

}

**Core/reducers/authReducers.ts**

import { Action, createReducer, on } from "@ngrx/store";

import { AuthType } from "../config/auth.types";

import \* as authAction from "../actions/auth.actions";

export const authKey = "auth";

export const initialState:AuthType = {

    isAuth: false,

    userName: "",

    roleDetails: {},

    accessPagesList:[]

};

const authReducer = createReducer(

    initialState,

    on(authAction.saveLoginAction, (state, {payload})=>{

        return {

            ...state,

            userName: payload

        }

    }),

    on(authAction.resetLoginAction, (state)=>{

        return {

            ...initialState

        }

    })

);

export function reducer(state: AuthType | undefined, action: Action){

    return authReducer(state, action)

}

**Step6 >>**

Create core folder in app module and in this module, create 3 folder as actions, effects and reducers with each index files.

**Core/actions/common.actions.ts**

import { createAction } from "@ngrx/store";

export const sendEmptyAction = createAction(

    '[common] send empty action'

);

**core/effects/index.ts**

import { AuthEffects } from './../../../auth/core/effects/auth.effects';

export const Effects = [

    AuthEffects

]

**Core/reducers/index.ts**

import { ActionReducer, ActionReducerMap, MetaReducer } from "@ngrx/store";

import { localStorageSync } from "ngrx-store-localstorage";

import \* as auth from "../../../auth/core/reducers/auth.reducers";

export const reducers:ActionReducerMap<any> = {

    auth: auth.reducer,

}

export function localStorageSyncReducer(

    reducer: ActionReducer<any>

): ActionReducer<any> {

    return localStorageSync({

        keys: [auth.authKey],

        rehydrate: true,

        storage: sessionStorage,

    })(reducer);

}

export function clearState(

    reducer: ActionReducer<any>

): ActionReducer<any> {

    return (state, action) => {

        if(action !== null && action.type === "RESET\_STATE"){

            return reducer(undefined, {type: "RESET\_STATE"});

        }

        return reducer(state, action);

    }

}

export const metaReducers: Array<MetaReducer<any,any>> = [

    localStorageSyncReducer,

    clearState

]