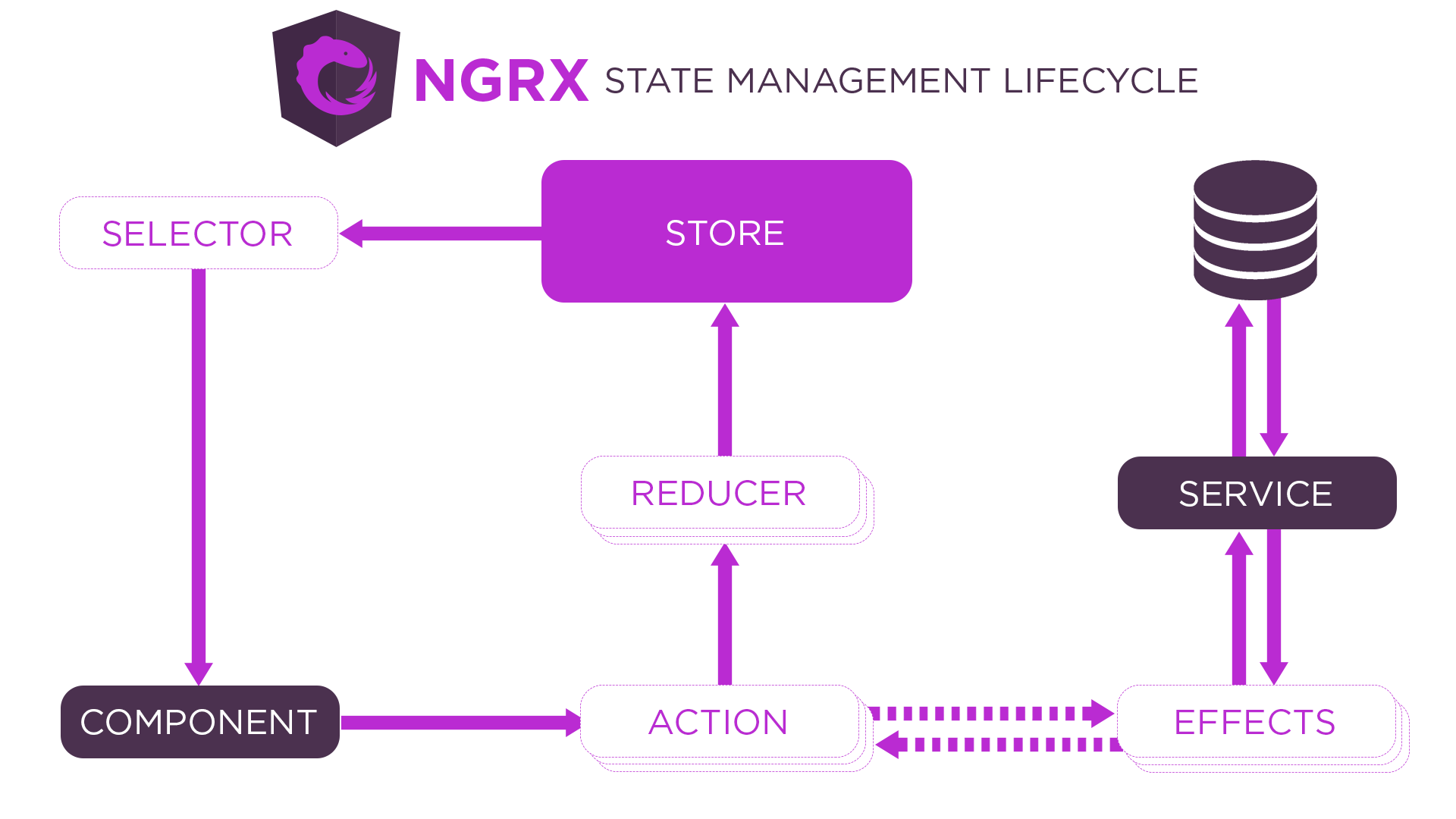
**How To Implement NGRX Store In Angular Apps**



**Introduction**

In this blog , we will learn, how we can implement ngrx store on our project.

First before start anything , we have to know, what is ngrx ???

NgRx Store is a state management library for Angular apps. In simple words, we can say like we have to manage data on client side with this library. With Angular apps, we can use states with data for single page applications.

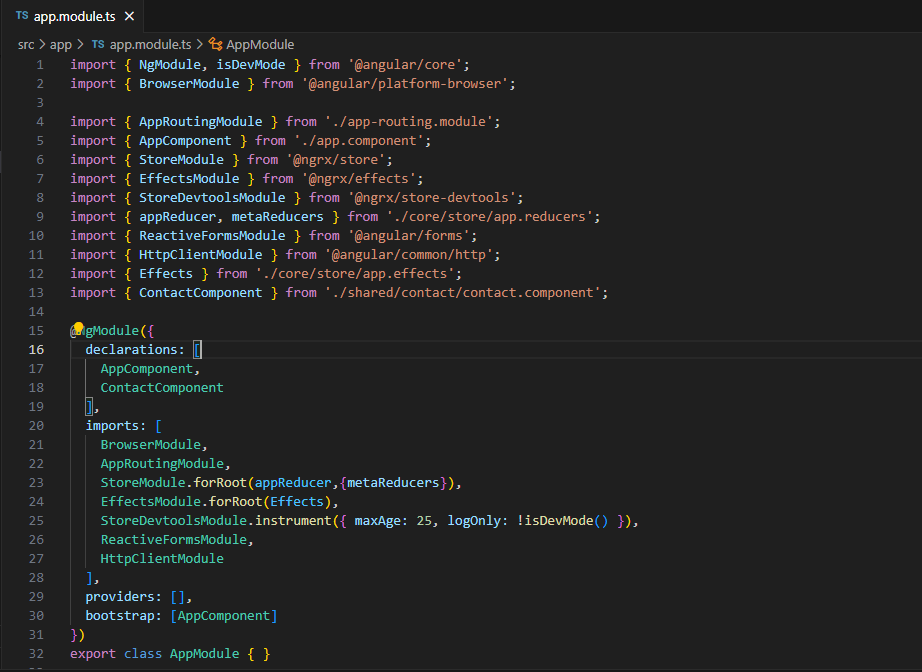
It makes Angular development easy by simplify the application’s state in objects with unidirectional data flow.

**Setup of NgRx Store**

1. You can install the Store to your project with the following ng add command.
2. ng add @ngrx/store
3. If you want to install by npm, then you can use below command
4. npm install @ngrx/store –save

Import the **StoreModule** to your **app.module.ts** and add **StoreModule.forRoot({}, {})** to array imports:

App.module.ts file link

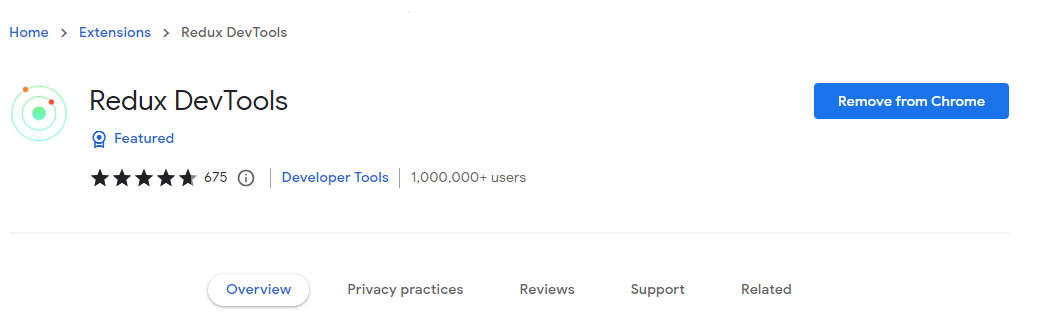


<https://ngrx.io/guide/store>

Now, you have to install **@ngrx/store-devtools** on your project for provide developer tool information, also we have to install it’s browser’s extension.

1. ng add **@ngrx/store-devtools@latest**
2. Update your **src/app.module.ts** > imports array **with StoreDevtoolsModule.instrument({ maxAge: 25, logOnly: !isDevMode() }).** The maxAge property will be set to the flag maxAge if provided.
3. If you want to add by npm then below command will help you
4. npm install @ngrx/store-devtools –save

Now we have to add Redux Devtools Extension for Google Chrome.

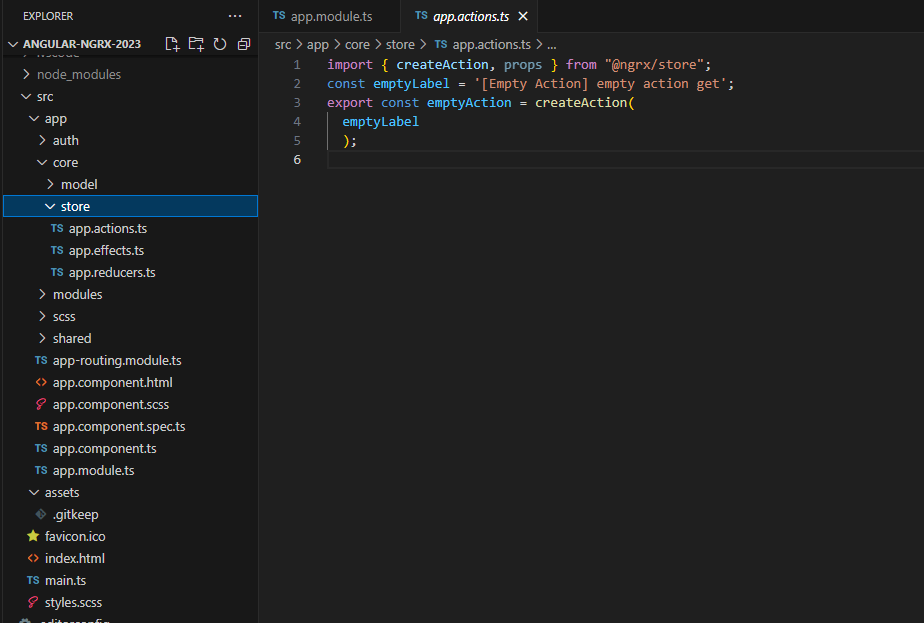


Now Install Effects

In a service-based Angular application, components are responsible for interacting with external resources directly through services. Instead, effects provide a way to interact with those services and isolate them from the components

1. ng add @ngrx/effects@latest
2. If you want to add by npm, please refer below command
3. npm install @ngrx/effects –save
4. Update your src/app/app.module.ts > imports array with EffectsModule.forRoot([AppEffects]).

**Create File Structure for Store Part**



This file structure, we created for root store part, we have to manage our state. I created here actions, effects and reducers.

**Action >>** Action describe all the changes in the state of the application means we have to call action for manage data to reducers and api services in effects.

**Reducers >>** It is a immedator between store and action that carry data for store as per call the action.

**Store >>** It is a central store that hold all of the application state in the application.

**Effects >>** For user services and api as per the action and manage those request/response for store as per action.

**App.action.ts**

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

const emptyLabel = '[Empty Action] empty action get';

export const emptyAction = createAction(

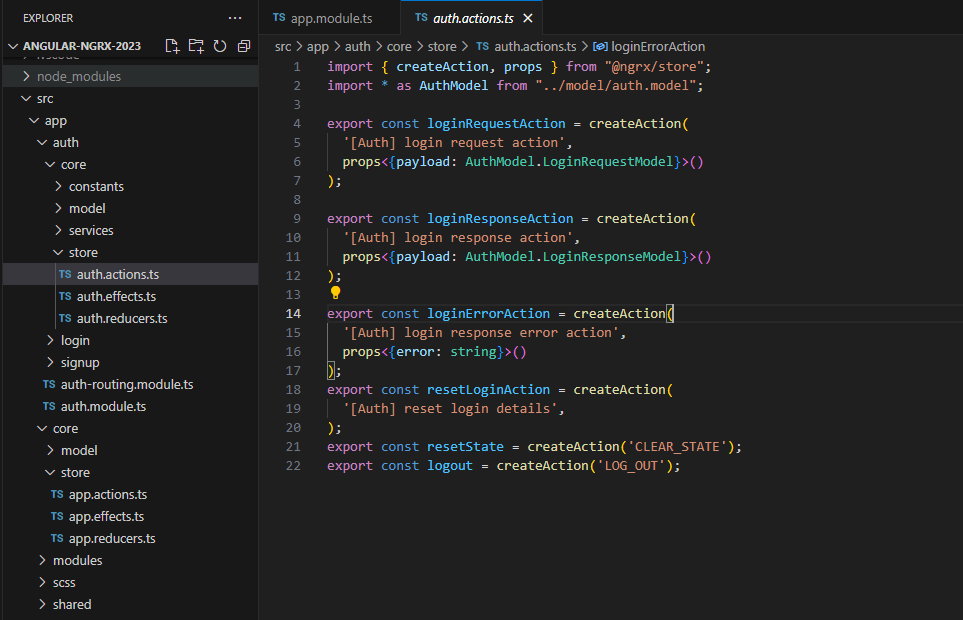
  emptyLabel

  );

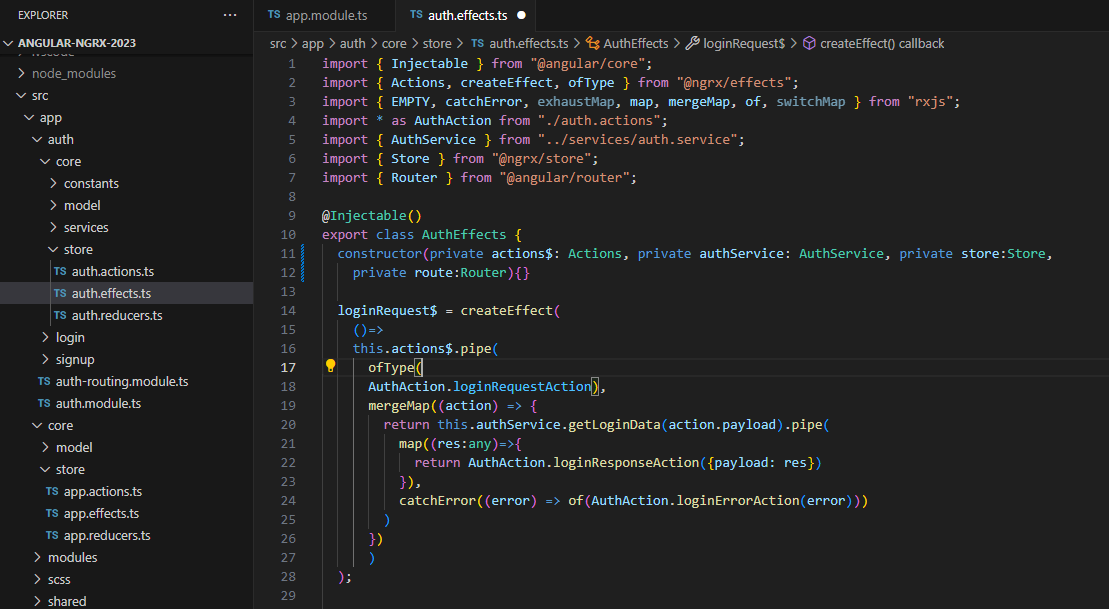
First we have to create action by **createAction()** method and to use it as per requirement.

Then whatever module by default , we have to take on load, I created auth module for store as by default.

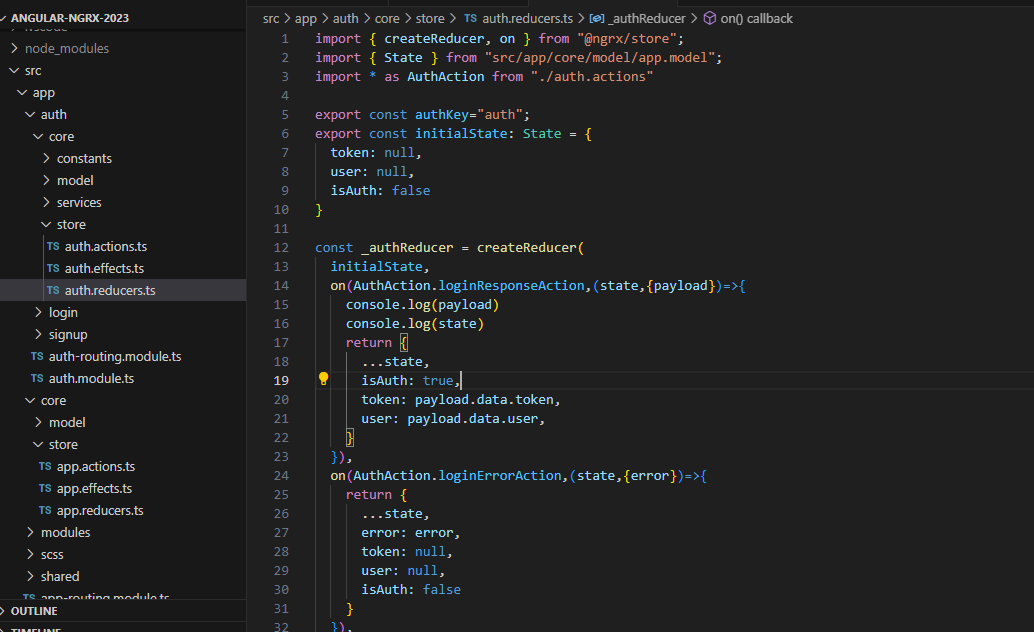
And create file structure with action, effects and reducers.



We have to create actions for different purpose for manage store as above image.

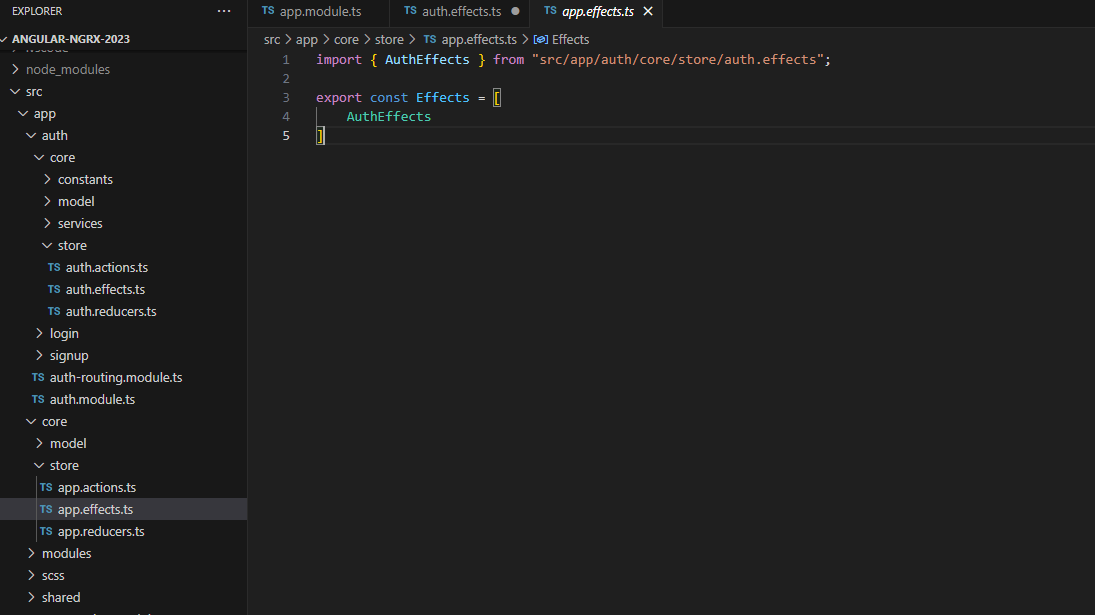


Here we are creating effects for calling API service for get data from service and send it to reducers by action for manage state in store.

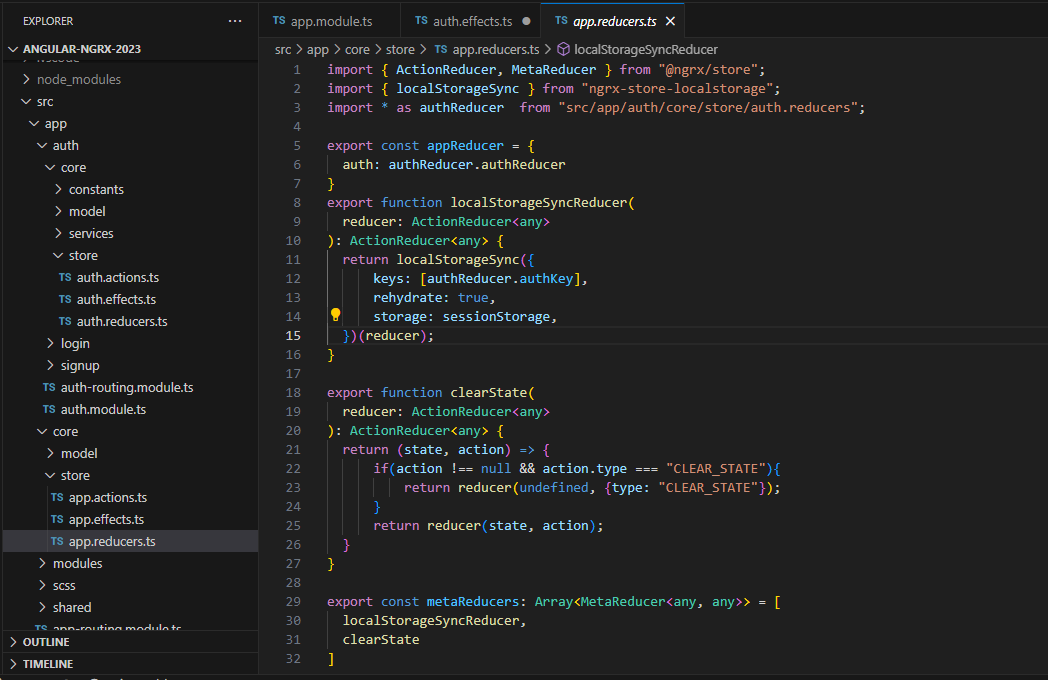


Here , we have to add reducers for manage state by action to store with **on()** method that should return state.

Now all these effects and reducers, we have to add on parent app part like this.

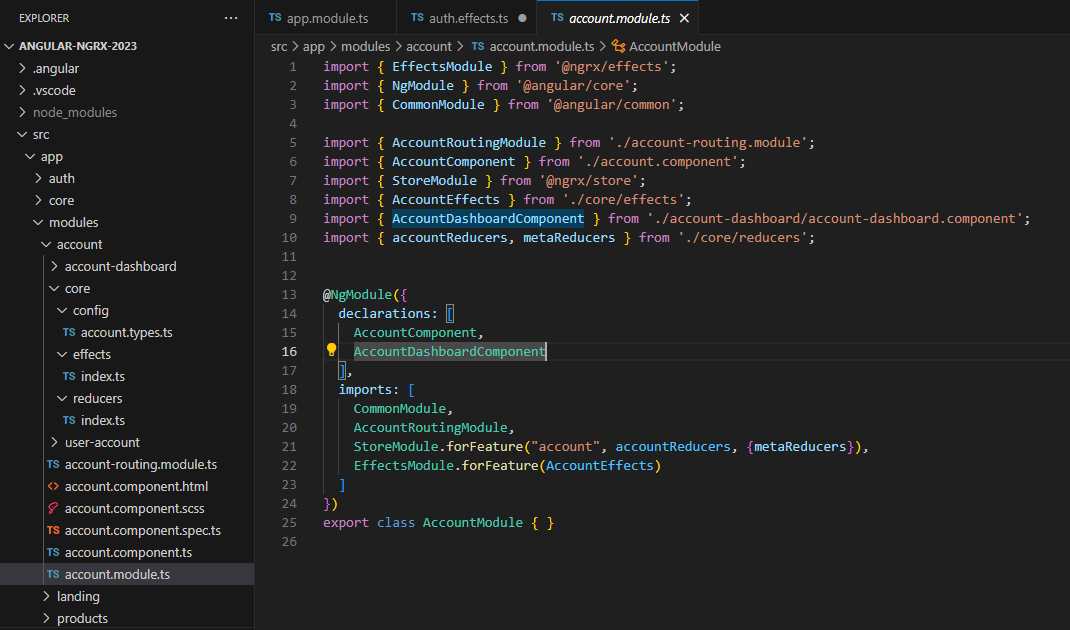


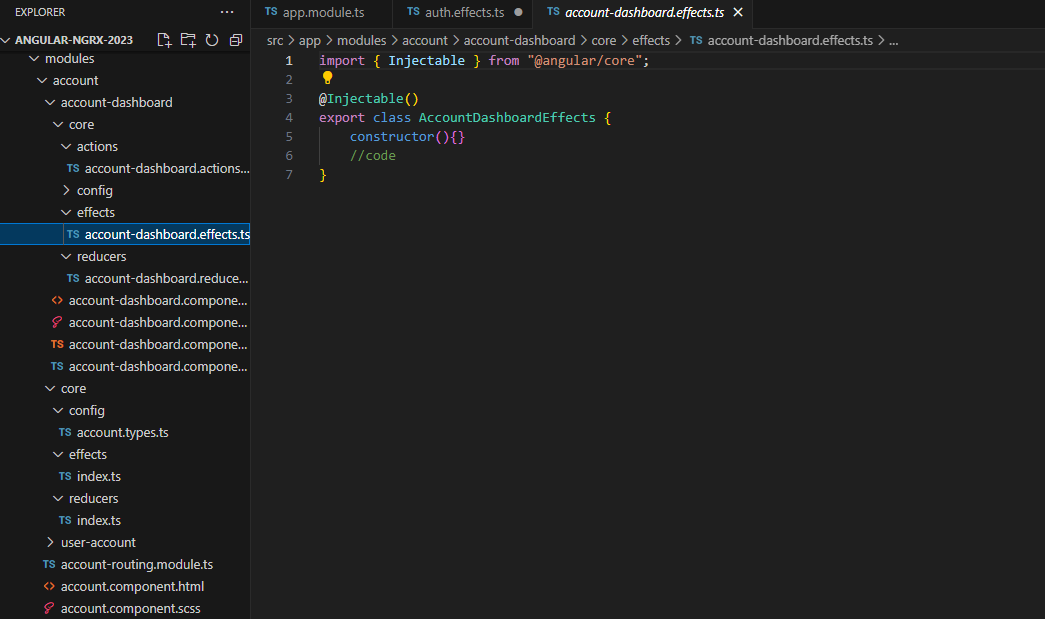
Here what effects , we added on particular module, then we have to export that effect here for call in main app.module.ts file

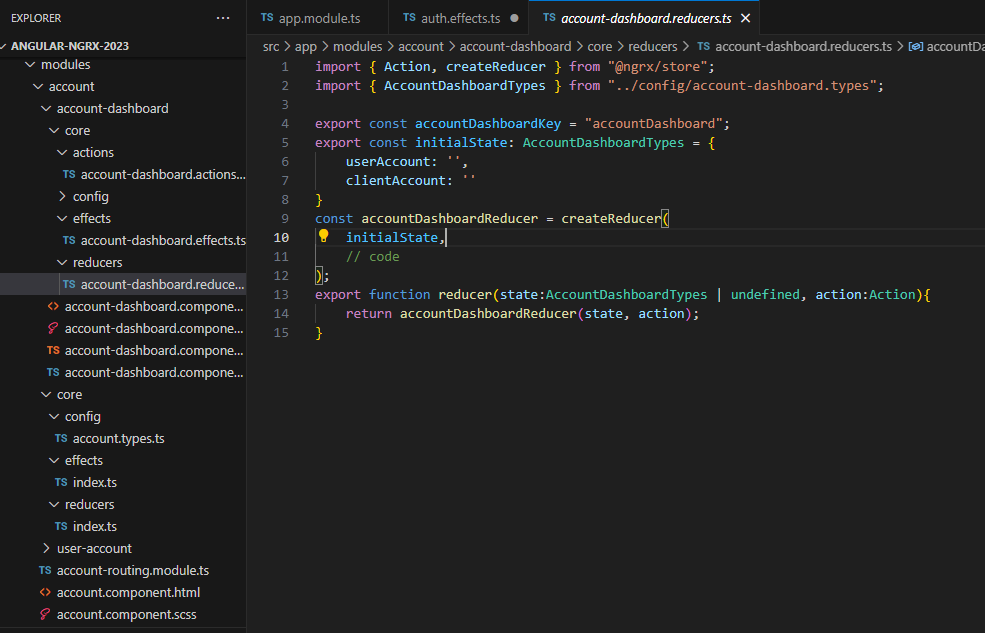


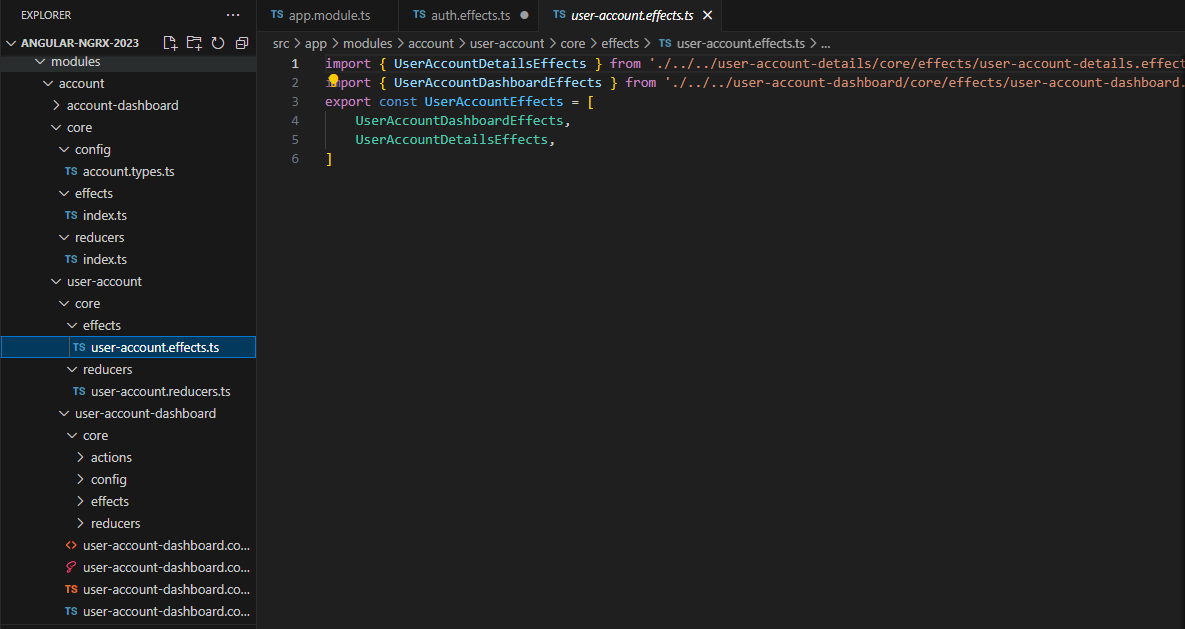
Here, which reducers, we wrote for manage data and state , we have to export hare for bind to main app module.

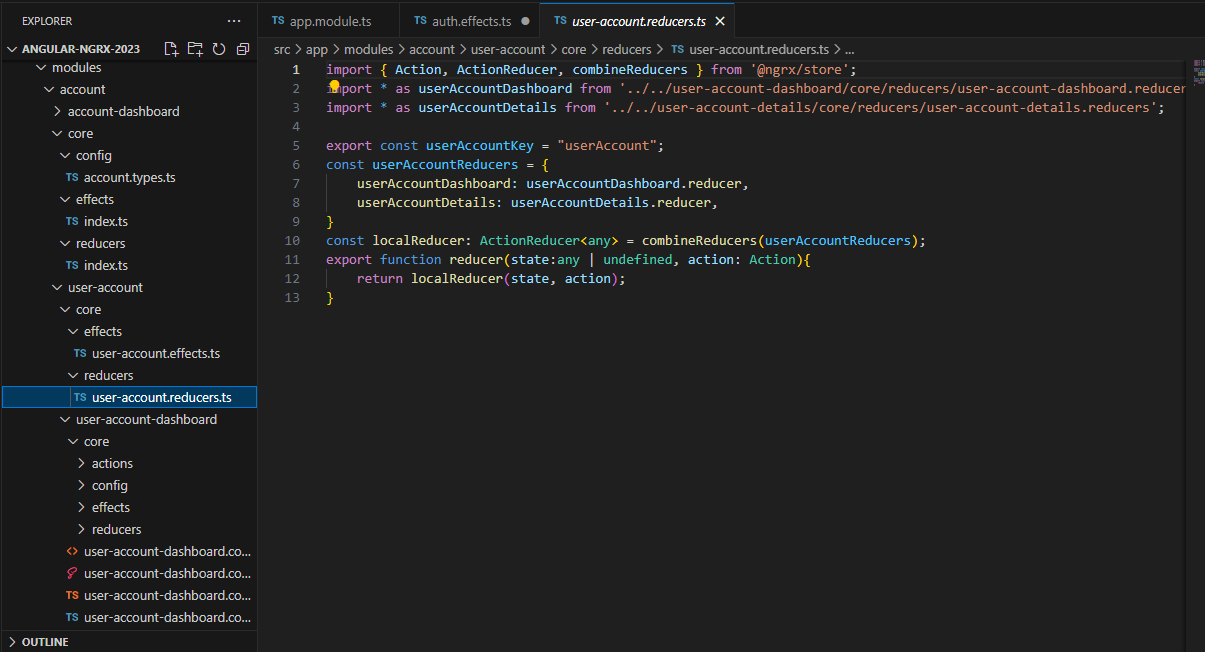
Now, If we want features modules with store, then first we have to create feature module and same file structure follow as app module.



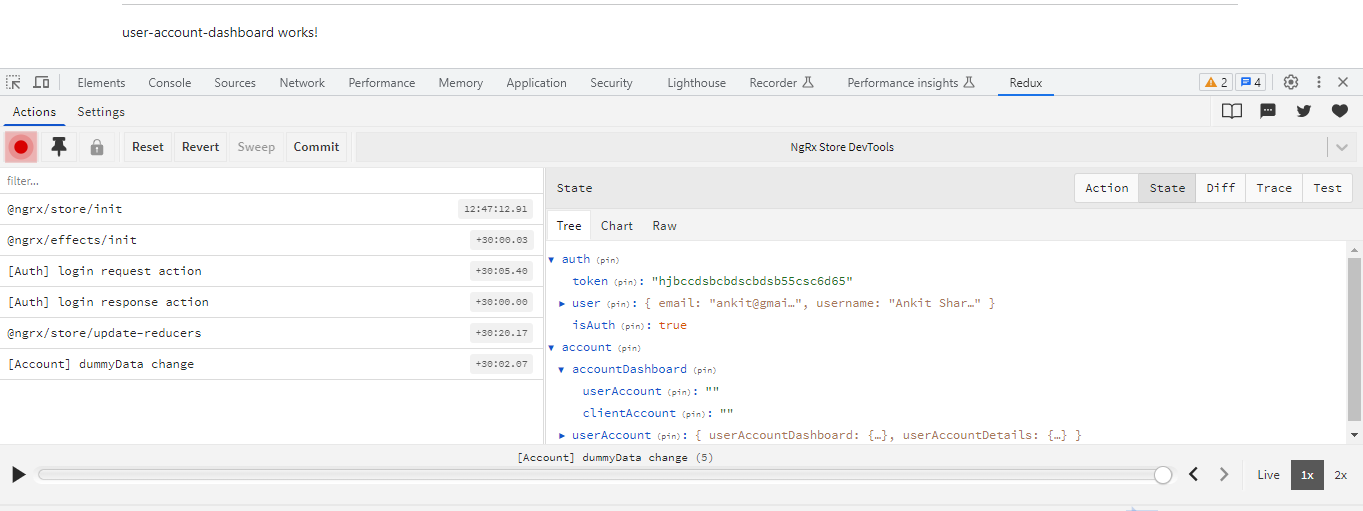








So , finally we implement NgRx in our application, but how we can debug these state , so whatever Redux devtools Extension, we install in starting, we have to go on that extension and open our inspect element part of google chrome and check this.



Here, we can check our state as different keys like **auth, account** and within these keys also manage different data.

This is the process , how we can setup and manage our state on angular project.

<https://github.com/ankitkumarsharma/angular-ngrx-2023>

<https://github.com/ankit_k_sharma>

[www.ankitkumarsharma.com](http://www.ankitkumarsharma.com)

Thanks