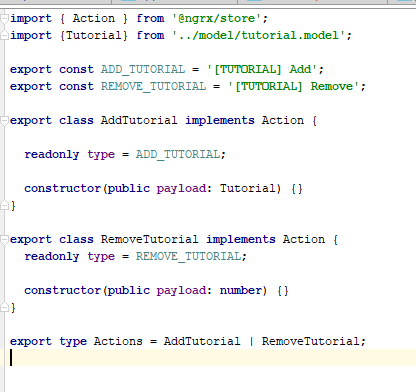
Ngrx Example

Action: An action in Ngrx/store is two things:

* A type in the form of a string. It describes what's happening.
* It contains an optional payload of data.

What is going to happen with it depends on the reducer.



Consts used as strings for the action type.

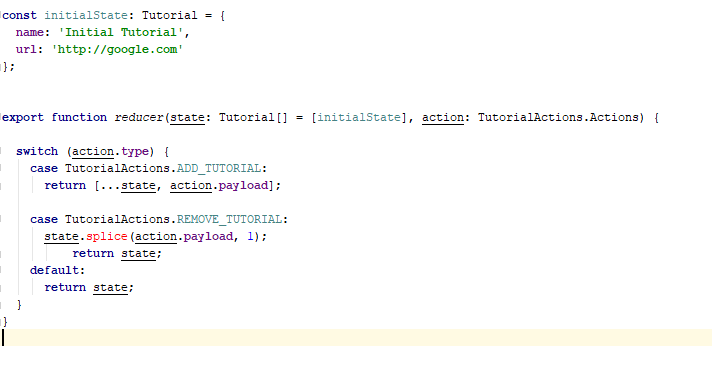
Export available actions which have to specify its type and a constructor (args are optional)

Export type actions then when imported in other file both actions will be available



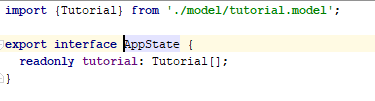


Reducer: A reducer is what takes the incoming action and decides what to do with it. It takes the previous state and returns a new state based on the given action.



Can have an initial state which will be return in case that reducer is selected before any action is executed.

State: state of the application, this will be return by the reducer and made available on the store.

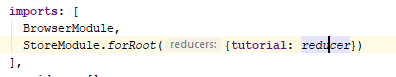


Store is injected to the component with generic type of the wanted state



Config:

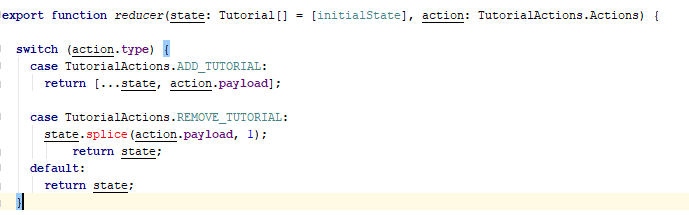




Import the reducers on app module. (name convention, tutorial.reducer.ts would be the file for the tutorial reducer added on the store module for root above.) tutorial should be selected on the store in order to call that reducer and get the reducer from the store:



Read from the store flow:



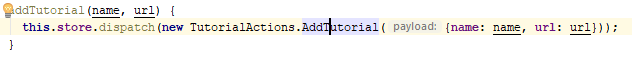
At start reducer is called, it will return the default value of the reducer (will send back the init value of the state)



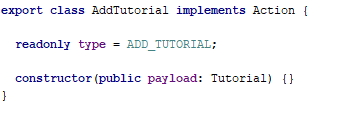
Component constructor calls the reducer tutorial with select, to retrieve info from the store.

Write to the store flow:

Component method called and it dispatch to the store



On the argument an new action is sent as argument, so it goes to actions to create that action type AddTutorial:



Then the reducer is called and will execute the according method depending on the action type:

