

Reducers

Reducers

Reducers sole function is to specify how the state / immutable Store changes in response to an Action. The reducer's job is to listen for all actions and then when it recognizes an action it performs a particular task that usually involves updating the immutable Store.

Every feature has its own `state` folder and corresponding `reducer.ts` file. In the reducer file you can find the definition of the State name, the selectors and the reducer itself, which is the definition of how the actions affect state. The reducer definition is a switch statement that lists the response to every defined action.

Example of a reducer.ts file:

```
1 import { User } from '../user';
2 import { createFeatureSelector, createSelector } from '@ngrx/store';
3 import { UserActions, UserActionTypes } from '../user.actions';
4
5 export interface UserState {
6   maskUserName: boolean;
7   currentUser: User;
8   currentUserAuthenticated: boolean;
9 }
10
11
12 const initialState: UserState = {
13   maskUserName: true,
14   currentUser: null,
15   currentUserAuthenticated: false
16 };
17
18 const getProductFeatureState = createFeatureSelector<UserState>('users');
19
20 export const getMaskUserName = createSelector(
21   getProductFeatureState,
22   state => state.maskUserName
23 );
24
25 export const getCurrentUser = createSelector(
26   getProductFeatureState,
27   state => state.currentUser
28 );
29
30 export const isCurrentUserAuthenticated = createSelector(
31   getProductFeatureState,
32   state => state.currentUserAuthenticated
33 );
34
35 export function reducer(state = initialState, action: UserActions): UserState {
36   switch (action.type) {
37
38     case UserActionTypes.ToggleMaskUserName:
39       return {
40         ...state,
41         maskUserName: action.payload
42       };
43
44     case UserActionTypes.SetCurrentUser:
45       return {
46         ...state,
47         currentUser: { ...action.payload }
48       };
49
50     case UserActionTypes.ClearCurrentUser:
51       return {
52         ...state,
```

```
export function reducer (
  state: ProductState = initialState,
  action: ProductActions ): ProductState {

  switch (action.type) {

    case ProductActionTypes.ToggleProductCode:
      return {
        ...state,
        showProductCode: action.payload
      };

    case ProductActionTypes.SetCurrentProduct:
      return {
        ...state,
        currentProduct: { ...action.payload }
      };

    case ProductActionTypes.InitializeCurrentProduct:
      return {
        ...state,
        currentProduct: {
          id: 0,
          productName: '',
          productCode: 'New',
          description: ''
        }
      };
    ....
  }
}
```