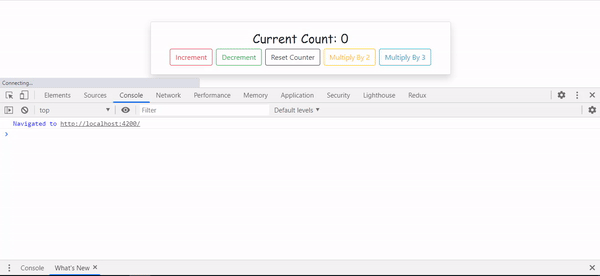
**Problem Statement:-**Log all the dispatched actions and state changes in browsers console as shown below:



Update the code in **app.module.ts**with the code given below in the counter-app used in previous demo:

1. import { BrowserModule } from '@angular/platform-browser';
2. import { NgModule } from '@angular/core';
3. import { AppRoutingModule } from './app-routing.module';
4. import { AppComponent } from './app.component';
5. import { CounterComponent } from './counter/counter.component';
6. import { StoreModule, ActionReducer, MetaReducer } from '@ngrx/store';
7. import { counterReducer } from './Store/reducers/counter.reducer';
8. */\* console.log all actions and state changes \*/*
9. export function debug(reducer: ActionReducer<any>): ActionReducer<any> {
10. return function (state, action) {
11. console.log('previous state', state);
12. console.log('action', action);
13. let nextState = reducer(state, action);
14. console.log('current state', nextState);
15. return nextState;
16. };
17. }
18. export const metaReducers: MetaReducer<any>[] = [debug];
19. @NgModule({
20. declarations: [
21. AppComponent,
22. CounterComponent
23. ],
24. imports: [
25. BrowserModule,
26. AppRoutingModule,
27. StoreModule.forRoot({ count: counterReducer }, { metaReducers })
28. ],
29. providers: [],
30. bootstrap: [AppComponent]
31. })
32. export class AppModule { }

**Line 7:** **ActionReducer**and **MetaReducers**are imported from @ngrx/store. **ActionReducer**is a function that takes an Action and a state and return another State. It is used to ensure, debug function takes a reducer function as a parameter.

**Line 12:** **debug** is a reducer function that takes a **ActionReducer** as a parameter and returns another **ActionReducer.**It returns another function that logs the dispatched action, state before and after the action was handled by the corresponding reducer.

**Line 22: metaReducer**is an array of meta-reducers to be used, in this case we are using only one i.e. **debug**meta-reducer.

**Line 32: metaReducer**has been configured to the application using **StoreModule.forRoot()**