§ NGRX Training Part 2 - Actions and Reducers Refined

Friday, August 14, 2020 6:52 PM

Introduction

ngrx/store provides class support that helps refine how actions, reducers, and selectors are defined. In this tutorial, we'll explore those refinements by updating the list management application from Part 1 of this tutorial.

Better Actions

ngrx/store provides a `createAction` function that can be used to handle state transitions.

```
createAction<T extends string, C extends Creator>(type: T, config?: C | { _as: "props"; }):
ActionCreator<T>
```

Parameters

type	Describes the action that will be dispatched.
config	Additional metadata needed for the handling of the action. Optional. Default is undefined.

From <https://ngrx.io/api/store/createAction>

We can utilize `createAction` to update the actions related to `iceCream`.

ice-cream-action.ts

```
import { createAction, props } from '@ngrx/store';
import { IceCream } from '../models/ice-cream';


// Use createAction to define a function that will produce a desired Action
// with the data that is defined on the Action.

export const AddFlavor = createAction('[ICECREAM] Add', props< {iceCream: IceCream} >());
export const RemoveFlavor = createAction('[ICECREAM] Remove', props< {flavorId: number} >());
```

Better Reducers

ngrx/store provides a `createReducer` function that can be used to update state.

```
createReducer<S, A extends Action = Action>(initialState: S, ...ons: On<S>[]):
ActionReducer<S, A>
```

Parameters

initialState	Provides a state value if the current state is undefined, as it is initially.
ons	Associations between actions and state changes.

From <https://ngrx.io/api/store/createReducer>

We can utilize `createReducer` as shown in the following update to the `ice-cream-reducer`.

ice-cream-reducer.ts

```
import { createReducer, on } from '@ngrx/store';
import { IceCream } from '../models/ice-cream';
```

```
import { AddFlavor, RemoveFlavor} from '../actions/ice-cream-actions';

const initialState: IceCream = {
    flavor: 'Vanilla',
    maker: 'Honey Hut'
}

export const iceCreamReducer = createReducer<IceCream[]>(
    [initialState],
    on(AddFlavor, (state, action) => {
        return [...state, action.iceCream];
    }),
    on(RemoveFlavor, (state, action) => {
        return state.filter((flavor, index , stateArray) => (index !== action.flavorId));
    })
);
```

Note the removal of the switch statement and its use of a string constant, and the syntax for initial state.