import {

Component, OnInit, OnDestroy, ViewChild} from '@angular/core';

import { Store } from '@ngrx/store';

import { Ingredient } from '../../shared/ingredient.model';

import \* as ShoppingListActions from '../store/shopping-list.actions';

import \* as fromShoppingList from '../store/shopping-list.reducers';

@Component({

selector: 'app-shopping-edit',

templateUrl: './shopping-edit.component.html',

styleUrls: ['./shopping-edit.component.css']

})

export class **ShoppingEditComponent implements OnInit, OnDestroy** {

@ViewChild('f') slForm: NgForm;

subscription: Subscription;

**editMode = false;**

**editedItem: Ingredient;**

constructor(**private store: Store<fromShoppingList.AppState>**) { }

ngOnInit() {

this.subscription = this.store.select('shoppingList')

.subscribe(

data => {

if (data.editedIngredientIndex > -1) {

this.editedItem = data.editedIngredient;

**this.editMode = true;**

this.slForm.setValue({

name: this.editedItem.name,

amount: this.editedItem.amount

})

} else {

this.editMode = false;}});}

**onSubmit(form: NgForm) {**

const value = form.value;

const newIngredient **= new Ingredient(value.name, value.amount);**

if (this.editMode) {

this.store.dispatch(new ShoppingListActions.**UpdateIngredient({ingredient: newIngredient}))**

} else {

**this.store.dispatch(new ShoppingListActions.AddIngredient(newIngredient))**

}

**this.editMode = false**;

form.reset();

}

onClear() {

this.slForm.reset();

this.editMode = false;

}

onDelete() {

**this.store.dispatch(new ShoppingListActions.DeleteIngredient());**

this.onClear();

}

ngOnDestroy() {

**this.store.dispatch(new ShoppingListActions.StopEdit());**

this.subscription.unsubscribe();

}

}

import { Ingredient } from '../shared/ingredient.model';

import \* as fromShoppingList from './store/shopping-list.reducers';

import \* as ShoppingListActions from './store/shopping-list.actions';

@**Component**({

selector: 'app-shopping-list',

**templateUrl**: './shopping-list.component.html',

**styleUrls**: ['./shopping-list.component.css']

})

export class **ShoppingListComponent** implements **OnInit** {

shoppingListState: Observable<{ingredients: Ingredient[]}>;

constructor(private store: **Store<fromShoppingList.AppState>**) { }

ngOnInit() {

this.shoppingListState = **this.store.select('shoppingList');**

}

onEditItem(index: number) {

**this.store.dispatch(new ShoppingListActions.StartEdit(index))**;

}

}------

export const **ADD\_INGREDIENT = 'ADD\_INGREDIENT'**;

export const **ADD\_INGREDIENTS = 'ADD\_INGREDIENTS';**

export const **UPDATE\_INGREDIENT = 'UPDATE\_INGREDIENT'**;

export const **DELETE\_INGREDIENT = 'DELETE\_INGREDIENT';**

export const **START\_EDIT = 'START\_EDIT';**

export const **STOP\_EDIT = 'STOP\_EDIT';**

export class AddIngredient implements Action {

readonly type = ADD\_INGREDIENT;

**constructor(public payload: Ingredient) {}}**

export class AddIngredients implements Action {

readonly type = ADD\_INGREDIENTS;

**constructor(public payload: Ingredient[]) {}}**

export class UpdateIngredient implements Action {

readonly type = UPDATE\_INGREDIENT;

**constructor(public payload: {ingredient: Ingredient}) {}}**

export class DeleteIngredient implements Action {

**readonly type = DELETE\_INGREDIENT;}**

export class StartEdit implements Action {

readonly type = START\_EDIT;

**constructor(public payload: number) {}**

}

export class StopEdit implements Action {

readonly type = STOP\_EDIT;}

export type ShoppingListActions =

AddIngredient |

AddIngredients |

UpdateIngredient |

DeleteIngredient |

StartEdit |

StopEdit;

import { Ingredient } from '../../shared/ingredient.model';

export **interface AppState** {

shoppingList: **State**}

export interface State {

**ingredients**: Ingredient[];

**editedIngredient**: Ingredient;

**editedIngredientIndex**: number;}

const **initialState: State** = {

ingredients: [

new Ingredient('Apples', 5),

new Ingredient('Tomatoes', 10),],

**editedIngredient**: null,

**editedIngredientIndex**: -1

};

export function **shoppingListReducer**(state = **initialState**, action: **ShoppingListActions**.**ShoppingListActions**) {

switch (action.type) {

case ShoppingListActions.**ADD\_INGREDIENT**:

return {

...**state**,

ingredients: [...state.ingredients, **action.payload**]

};

case ShoppingListActions**.ADD\_INGREDIENTS**:

return { ...state,

ingredients: [...**state.ingredients**, **...action.payload**]

};

case ShoppingListActions.**UPDATE\_INGREDIENT**:

const ingredient = **state.ingredients[state.editedIngredientIndex];**

const updatedIngredient = {

...ingredient,

...**action.payload.ingredient**

};

const ingredients = [...state.**ingredients**];

ingredients[state.**editedIngredientIndex**] = **updatedIngredient**;

return {...state,

ingredients: **ingredients**,

editedIngredient: null, **editedIngredientIndex**: -1

};

case ShoppingListActions.DELETE\_INGREDIENT:

const **oldIngredients** = [...state.ingredients];

**oldIngredients**.**splice**(state.**editedIngredientIndex**, 1);

return {...state,

ingredients: **oldIngredients**,

editedIngredient: null, **editedIngredientIndex**: -1

};

case ShoppingListActions.START\_EDIT:

const editedIngredient = {...**state.ingredients[action.payload**]};

return { ...state,

editedIngredient: **editedIngredient**,

**editedIngredientIndex**: **action.payload**

};

case ShoppingListActions.**STOP**\_**EDIT**:

return { ...state,

editedIngredient: null, **editedIngredientIndex**: -1

};

default:

return **state**;}}

import { Component, OnInit } from '@angular/core';

import { ActivatedRoute, Params, Router } from '@angular/router';

import { Store } from '@ngrx/store';

import { Recipe } from '../recipe.model';

import { RecipeService } from '../recipe.service';

import \* as ShoppingListActions from '../../shopping-list/store/shopping-list.actions';

import \* as fromShoppingList from '../../shopping-list/store/shopping-list.reducers';

@**Component**({

**selector: 'app-recipe-detail'**,

templateUrl: './recipe-detail.component.html',

styleUrls: ['./recipe-detail.component.css']

})

export class RecipeDetailComponent implements OnInit {

recipe: Recipe;

id: number;

constructor(private recipeService: RecipeService,

private route: ActivatedRoute,

private router: Router,

private store: Store<fromShoppingList.AppState>) {

}

**ngOnInit**() {

this.route.params

.subscribe(

(params: Params) => {

this.id = +params['id'];

this.recipe = this.recipeService.getRecipe(this.id);

}

);

}

onAddToShoppingList() {

this**.store.dispatch**(new **ShoppingListActions.AddIngredients(this.recipe.ingredients**));

}

onEditRecipe() {

this.router.navigate(['edit'], {relativeTo: this.route});

// this.router.navigate(['../', this.id, 'edit'], {relativeTo: this.route});

}

onDeleteRecipe() {

this.recipeService.deleteRecipe(this.id);

this.router.navigate(['/recipes']);

}

}