Implementando o carrinho de compras:

Para parte de carrinho de compras, foi criado um serviço. Pois pode ser reutilizado:

import { CartItem } from './cart-item.model';

import { Menu } from '../menu/menu-model';

import { ThrowStmt } from '@angular/compiler/src/output/output\_ast';

export class ShoppingCartservice{

    items: CartItem[] = [];

    clear(){

            this.items = [];

    }

    addItem(item: Menu){

        //verificando se o item já existe

        let foundItem = this.items.find(mItem => mItem.menuItem.id == item.id);

        //se sim, adicionamos mais um na sua quantidade

        if(foundItem){

           foundItem.quantity ++;

         //se não, adicionamos o item ao array;

        }else{

Instanciado um novo CartItem e passando o item ao parâmetro menuItem do construtor.

            this.items.push(new CartItem(item))

        }

    }

    removeItem(item: CartItem){

        this.items.splice(this.items.indexOf(item), 1)

    }

    total(): number{

        return this.items

        //trocando o array apenas pelo valores

        .map(item => item.value())

        //executa uma função para cada membro do array, resultando em um único valor de retorno

        //nesse caso está fazendo a soma do total de todos valores do array, e seu valor inicial é zero.

        .reduce((prev, value)=> prev+value, 0);

    }

}

Repare que o array item, é do tipo cartItem, que é uma classe:

import { Menu } from '../menu/menu-model';

export class CartItem{

    constructor(public menuItem: Menu,

                public quantity: number = 1){

                }

                value(): number {

                    return this.menuItem.price \* this.quantity;

                }

}

Utilizando os métodos no nosso componente do carrinho:

 items(): any[]{

    return this.shoppingCartSrvc.items;

  }

  total(): number{

    return this.shoppingCartSrvc.total();

  }

removeItem(param){

  console.log('evento', param)

  this.shoppingCartSrvc.removeItem(param);

}

addItem(param){

  this.shoppingCartSrvc.addItem(param);

}

  clear(){

    this.shoppingCartSrvc.clear();

  }

O link adicionar item fica no componente de menu, e para fazer com que ele chamasse o método que se encontra no componente do carrinho, usamos template variable, ex:

 <mt-menu-item \*ngFor="let item of menu"

  [menuItem]="item" (add)="shoppingCart.addItem($event)"></mt-menu-item>

Dessa Forma de outro componente, o output add irá receber o método addItem do componente de carrinho.

Lembrando que add é um eventEmiter no output do componente menuItem.