

Module 2 – Databinding and Animations

Demo 2: Bind data to shopping list component

edureka!

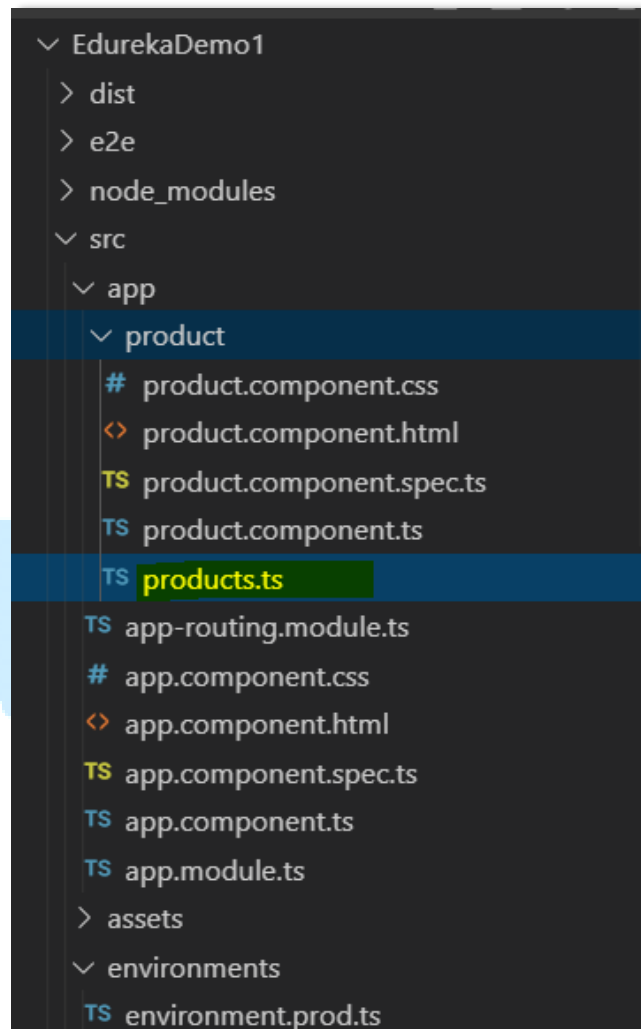
edureka!

© Brain4ce Education Solutions Pvt. Ltd.

In this demo, we will see how to bind data to shopping list component created in Module 2 demo 2

Step 1 – Open visual studio code, Open project folder created in Module 2 Demo2

Step 2 – Navigate to src → app → product folder, right click on product folder and select 'New File'. Name the new file as 'product.ts'



Product.ts is model file created for databinding

Step3 – Add the 5 properties/variables to product.ts as shown below

```
TS products.ts ×
EdurekaDemo1 > src > app > product > TS products.ts > Product
1  export interface Product {
2
3      p_id : number,
4
5      product_name : string,
6
7      product_weight : string,
8
9      product_price : number,
10
11     product_image :string
12
13 }
```

Step 4 – Open product.component.ts file to add the business logic. Create different arrays to store data.

```
TS product.component.ts ×
EdurekaDemo1 > src > app > product > TS product.component.ts > ProductComponent
11  templateUrl: './product.component.html',
12  styleUrls: ['./product.component.css']
13  })
14
15  export class ProductComponent implements OnInit {
16
17
18
19
20
21  products: Product[];
22
23  selectedProduct:Product;
24
25  productList: Product[]=[];
26
27  errorItem: boolean = false;
28
29  |
30
31  constructor() {}
32
33 }
```

Step 5 – Add sample products to this.products array, this.products array will be used to display product list.

```

TS product.component.ts
EdurekaDemo1 > src > app > product > TS product.component.ts > ProductComponent > constructor
30
31 constructor() {}
32
33
34 ngOnInit() {
35
36   this.products = [
37
38     {
39       p_id : 1,
40       product_name : 'Surf Excel',
41       product_weight : '1Kg',
42       product_price : 590,
43       product_image : 'https://rukminim1.flixcart.com/image/832/832/jg6v24w0/washing-powder/3/r/d/2-2-kg-top-load-surf-excel-orig
44     },
45     {
46       p_id : 2,
47       product_name : 'Nivea',
48       product_weight : '200ml',
49     }
50   ]
51 }
52
53
54
55
56
57
58
59

```

Step 6 – Create functions/methods to add and remove product operations

```

TS product.component.ts
EdurekaDemo1 > src > app > product > TS product.component.ts > ProductComponent > AddToCart
170
171
172
173 total() {
174   return this.productList.reduce((total, item) => total + item.product_price, 0);
175 }
176
177
178
179 addToCart(item) {
180   if(this.productList.indexOf(item) === -1) {
181     this.productList.push(item);
182     this.errorItem = false;
183   }
184   else if (this.productList.indexOf(item) > -1) {
185     this.errorItem = true;
186   }
187 }
188
189
190
191
192
193
194
195
196
197 removeItem(item) {
198   var index = this.productList.indexOf(item);
199

```

Step 7 – Open product.component.html file, modify html file to bind data to different html element

```

17
18
19 <div class="col-xl-4 col-sm-6 col-12 border border-warning mb-2" *ngFor = "let item of products">
20
21
22
23
24
25 <div class="row text-center mb-2">
26
27 <div class="col-12">
28 | 
29
30 </div>
31
32 </div>
33
34
35
36
37 <div class="row">
38
39 <div class="col-12">
40
41 | <h6 class="card-title font-weight-bold" > {{item.product_name}}</h6>
42
43 </div>
44
45 </div>
46

```

Step 8 – Open product.component.html file and bind data to the template, use *String Interpolation* for this. String interpolation is one-way databinding technique which is used to output the data from a TypeScript code to HTML template (view). It uses the template expression in double curly braces to display the data from the component to the view. The below screenshot shows string interpolation binding.

```

33 </div>
34
35
36
37 <div class="row">
38
39 <div class="col-12">
40
41 | <h6 class="card-title font-weight-bold" > {{item.product_name}}</h6>
42
43 </div>
44
45 </div>
46
47
48
49 <div class="row">
50
51 <div class="col-md-6">
52
53 | <p class="card-text"> <small> {{item.product_weight}} </small></p>
54
55 </div>
56
57 <div class="col-md-6">
58
59 | <p class="text-right"> <small> Rs {{item.product_price}} </small></p>
60
61 </div>
62
63 </div>
64

```

Step 9 – Bind data to src property of img html tag, we use *Property Binding* for this. In property binding, we bind a property of a DOM element to a field which is a defined property in our component TypeScript code.

```
<div class="row text-center mb-2">
  <div class="col-12">
    
  </div>
</div>
```

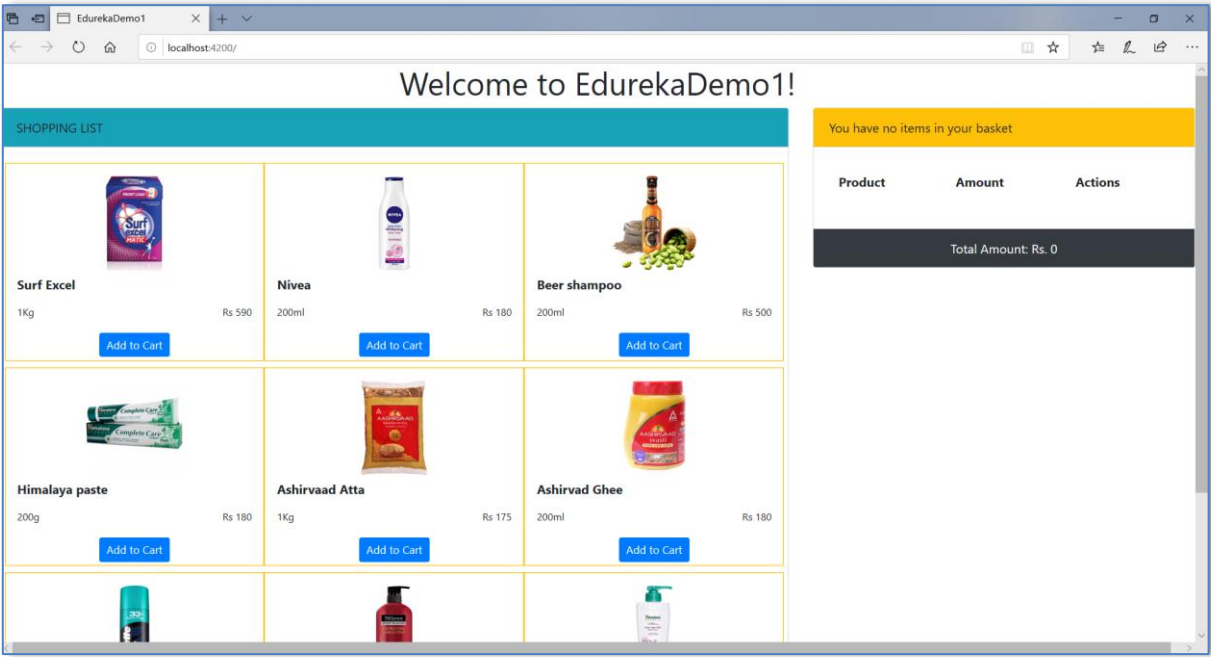
Step 10 – Bind to a click event and assign the event handler “addToCart(item)” to this event, use *Event Binding* for this. Event binding is used to handle the events raised from the DOM like button click, mouse move etc. When the DOM event happens (e.g. click, change), it calls the specified method in the component.

```
<div class="row text-center mb-1">
  <div class="col-12">
    <button class="btn btn-sm btn-primary" (click)="addToCart(item)">Add to Cart</button>
  </div>
</div>
```

Step 11 – Run angular application by using command ‘ng serve --open’

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS D:\Work\TapChief\Edureka\Angular8Demo\Module2\Demo1\edurekademo1> ng serve --open
```

Step 9 – This will open our shopping list app with data binding to html elements in the browser as shown below



Click on Add to cart to add product to cart, also remove button from cart will remove products from cart. Total amount will show sum of total amount of products added to cart.

edureka!