Data Binding in Angular - Demo

**Step 1:**

Open QuickKartApp folder in Visual Studio.

**Step 2:**

You will now modify the ViewProductsComponent to include the product details in the products array.

Open view-products.component.ts present in QuickKartApp --> src --> app --> view-products folder.

Add the following lines of code from 17 - 27 to add product details in products array.

1. import { Component, OnInit } from '@angular/core';
2. @Component({
3. selector: 'app-view-products',
4. templateUrl: './view-products.component.html',
5. styleUrls: ['./view-products.component.css']
6. })
7. export class ViewProductsComponent implements OnInit {
8. products: any[];
9. showMsgDiv: boolean = false;
10. constructor() { }
11. ngOnInit() {
12. this.products = [
13. { "productId": "P101", "productName": "Lamborghini Gallardo Spyder", "categoryId": 1, "price": 18000000, "quantityAvailable": 10 },
14. { "productId": "P102", "productName": "Ben Sherman Mens Necktie Silk Tie", "categoryId": 2, "price": 1847, "quantityAvailable": 20 },
15. { "productId": "P103", "productName": "BMW Z4", "categoryId": 1, "price": 6890000, "quantityAvailable": 10 },
16. { "productId": "P104", "productName": "Samsung Galaxy S4", "categoryId": 3, "price": 38800, "quantityAvailable": 100 }
17. ]
18. if (this.products == null)
19. {
20. this.showMsgDiv = true;
21. }
22. }
23. }

Here you have hard coded the values in the products array. You will learn to fetch it from the database later.

**Interpolation:**

**Step 3:**

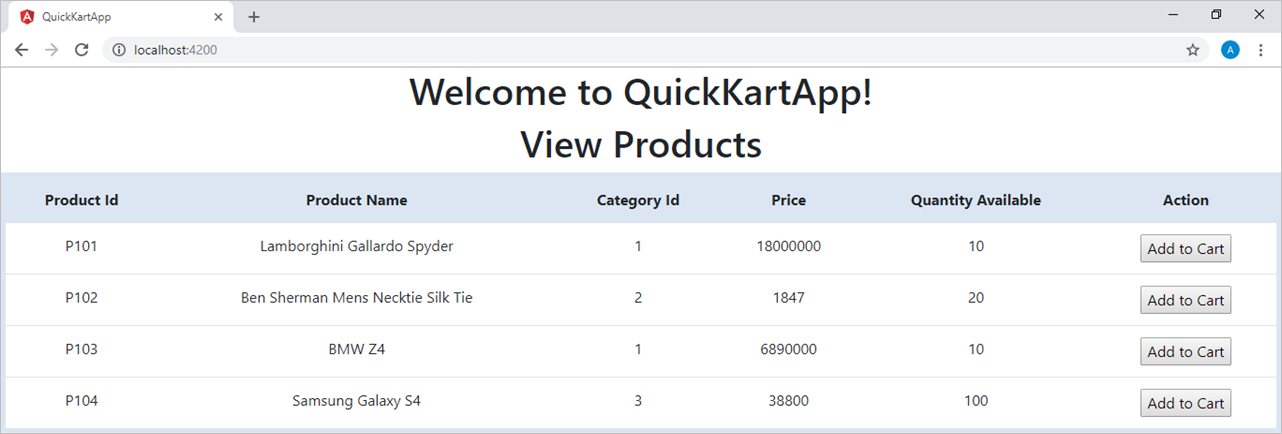
Open view-products.component.html and add the following code from line 14 - 19 to display the details of the products stored in products array.

1. <div style="text-align:center;">
2. <h1>View Products</h1>
3. <div class="table-responsive">
4. <table class="table" style="border:5px solid rgba(220,230,242,1);" \*ngIf="products">
5. <tr style="background-color:rgba(220,230,242,1); font-size:12pt">
6. <th>Product Id</th>
7. <th>Product Name</th>
8. <th>Category Id</th>
9. <th>Price</th>
10. <th>Quantity Available</th>
11. <th>Action</th>
12. </tr>
13. <tr \*ngFor="let product of products" style="background-color:white; font-size:10pt">
14. <td>{{product.productId }}</td>
15. <td>{{product.productName}}</td>
16. <td>{{product.categoryId}}</td>
17. <td>{{product.price}}</td>
18. <td>{{product.quantityAvailable}}</td>
19. <td><button title="Add to cart">Add to Cart</button></td>
20. </tr>
21. </table>
22. </div>
23. <div \*ngIf="showMsgDiv" style="">
24. <h4 class="jumbotron" style="text-align:center;">No products available</h4>
25. </div>
26. </div>

Observe that the product details are displayed with help of interpolation **{{product.property}}** which is one of the techniques of data binding in angular. A button is added so that visitors of the application can add the products to their cart.

**Step 4:**

Execute your application and check if the page is loaded as shown below.



Here the product details in the products array are not bound to any template and hence they are not strongly typed. So if there are any typographical errors, the application will not identify it before execution.

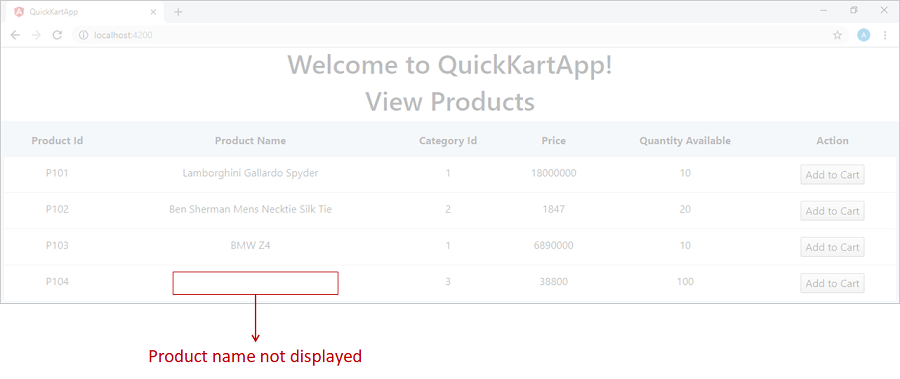
**Step 5:**

Go back to view-products.component.ts, introduce a typographical error in products array. For example, modify **productName**property of fourth product to **prodName**in line 5as shown below:

1. this.products = [
2. { "productId": "P101", "productName": "Lamborghini Gallardo Spyder", "categoryId": 1, "price": 18000000, "quantityAvailable": 10 },
3. { "productId": "P102", "productName": "Ben Sherman Mens Necktie Silk Tie", "categoryId": 2, "price": 1847, "quantityAvailable": 20 },
4. { "productId": "P103", "productName": "BMW Z4", "categoryId": 1, "price": 6890000, "quantityAvailable": 10 },
5. { "productId": "P104", "prodName": "Samsung Galaxy S4", "categoryId": 3, "price": 38800, "quantityAvailable": 100 }
6. ]

**Step 6:**

Observe that no error is detected while coding and in the browser the name of product with product id **P104**is not displayed.



Why was the product name not displayed?

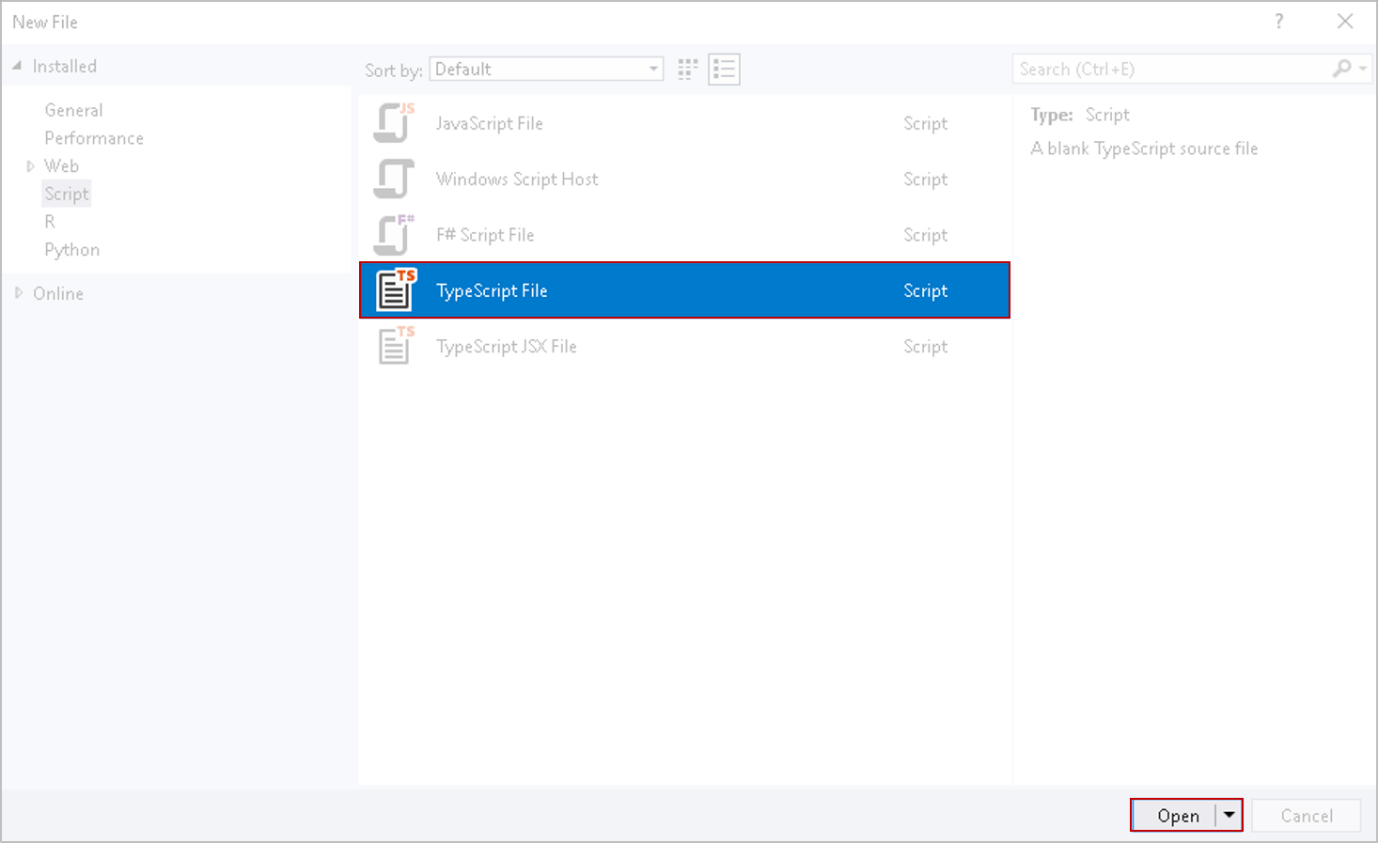
It was not displayed because the products array declared in the ViewProductsComponent class is not strongly typed i.e. it is declared of type **any** or it does not have a predefined type which negates the benefits of strong typing.

To make it strongly typed you can use a Product class or interface as the datatype of products array. As you need a template only to display data and no logic is required, you can make use of an interface.

**Step 7:**

In order to add an interface, right click on the app folder and select Add --> New Folder. Rename this folder as **quickKart-interfaces**.

Right click on quickKart-interfaces folder and select Add --> Add New Item --> Typescript file. Rename it as **product.ts**.



**Step 8:**

Open product.ts and add the following lines of code in it to create **IProduct**interface:

1. export interface IProduct {
2. productId: string;
3. productName: string;
4. price: number;
5. quantityAvailable: number;
6. categoryId: number;
7. }

**Step 9:**

Now import the IProduct in view-products.component.ts in line 2 as shown below.

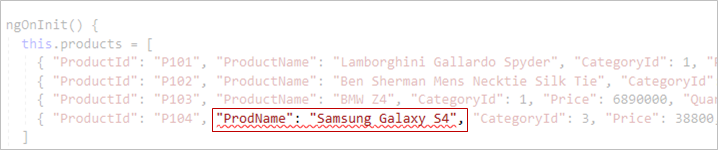
1. import { Component, OnInit } from '@angular/core';
2. import { IProduct } from '../quickKart-interfaces/product';
3. @Component({
4. selector: 'app-view-products',
5. templateUrl: './view-products.component.html',
6. styleUrls: ['./view-products.component.css']
7. })
8. export class ViewProductsComponent implements OnInit {
9. products: any[];
10. showMsgDiv: boolean = false;
11. constructor() { }
12. ngOnInit() {
13. this.products = [
14. { "productId": "P101", "productName": "Lamborghini Gallardo Spyder", "categoryId": 1, "price": 18000000, "quantityAvailable": 10 },
15. { "productId": "P102", "productName": "Ben Sherman Mens Necktie Silk Tie", "categoryId": 2, "price": 1847, "quantityAvailable": 20 },
16. { "productId": "P103", "productName": "BMW Z4", "categoryId": 1, "price": 6890000, "quantityAvailable": 10 },
17. { "productId": "P104", "productName": "Samsung Galaxy S4", "categoryId": 3, "price": 38800, "quantityAvailable": 100 }
18. ]
19. if (this.products == null)
20. {
21. this.showMsgDiv = true;
22. }
23. }
24. }

Here '..' refers to the parent folder of the current file.

Declare the products array of type IProduct in line 11.

1. import { Component, OnInit } from '@angular/core';
2. import { IProduct } from '../quickKart-interfaces/product';
3. @Component({
4. selector: 'app-view-products',
5. templateUrl: './view-products.component.html',
6. styleUrls: ['./view-products.component.css']
7. })
8. export class ViewProductsComponent implements OnInit {
9. products: IProduct[];
10. showMsgDiv: boolean = false;
11. constructor() { }
12. ngOnInit() {
13. this.products = [
14. { "productId": "P101", "productName": "Lamborghini Gallardo Spyder", "categoryId": 1, "price": 18000000, "quantityAvailable": 10 },
15. { "productId": "P102", "productName": "Ben Sherman Mens Necktie Silk Tie", "categoryId": 2, "price": 1847, "quantityAvailable": 20 },
16. { "productId": "P103", "productName": "BMW Z4", "categoryId": 1, "price": 6890000, "quantityAvailable": 10 },
17. { "productId": "P104", "prodName": "Samsung Galaxy S4", "categoryId": 3, "price": 38800, "quantityAvailable": 100 }
18. ]
19. if (this.products == null)
20. {
21. this.showMsgDiv = true;
22. }
23. }
24. }

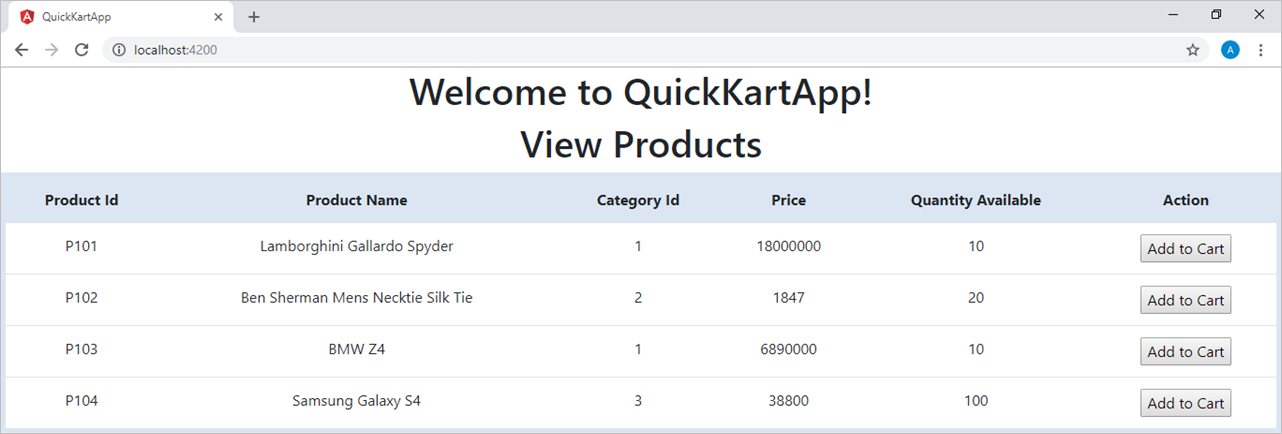
You can observe the syntax error as soon as **any**is replaced by **IProduct,**because now products array is strongly typed.



**Step 10:**

To fix the error, write **ProductName** instead of ProdName. You can also observe the intellisense  support for the product properties which is another great benefit of strong typing.

Now execute your application and check if the page is loaded as shown below.



**Event binding:**

You will now learn event binding by adding a category dropdownlist and displaying the product details based on the selected category.

**Step 11:**

Right click on quickKart-interfaces folder and select Add --> Add New Item --> Typescript file. Rename it as **category.ts**.

**Step 12:**

Open category.ts and add the following code in it to create **ICategory**interface.

1. export interface ICategory {
2. categoryId: number;
3. categoryName: string
4. }

**Step 13:**

Now import the ICategory in view-products.component.ts in line 3 as shown below:

1. import { Component, OnInit } from '@angular/core';
2. import { IProduct } from '../quickKart-interfaces/product';
3. import { ICategory } from '../quickKart-interfaces/category';
4. @Component({
5. selector: 'app-view-products',
6. templateUrl: './view-products.component.html',
7. styleUrls: ['./view-products.component.css']
8. })
9. export class ViewProductsComponent implements OnInit {
10. products: IProduct[];
11. showMsgDiv: boolean = false;
12. constructor() { }
13. ngOnInit() {
14. this.products = [
15. { "productId": "P101", "productName": "Lamborghini Gallardo Spyder", "categoryId": 1, "price": 18000000, "quantityAvailable": 10 },
16. { "productId": "P102", "productName": "Ben Sherman Mens Necktie Silk Tie", "categoryId": 2, "price": 1847, "quantityAvailable": 20 },
17. { "productId": "P103", "productName": "BMW Z4", "categoryId": 1, "price": 6890000, "quantityAvailable": 10 },
18. { "productId": "P104", "productName": "Samsung Galaxy S4", "categoryId": 3, "price": 38800, "quantityAvailable": 100 }
19. ]
20. if (this.products == null)
21. {
22. this.showMsgDiv = true;
23. }
24. }
25. }

**Step 14:**

Declare an array named **categories** of type ICategory at class level in view-products.component.ts in line 13.

1. import { Component, OnInit } from '@angular/core';
2. import { IProduct } from '../quickKart-interfaces/product';
3. import { ICategory } from '../quickKart-interfaces/category';
4. @Component({
5. selector: 'app-view-products',
6. templateUrl: './view-products.component.html',
7. styleUrls: ['./view-products.component.css']
8. })
9. export class ViewProductsComponent implements OnInit {
10. products: IProduct[];
11. categories: ICategory[];
12. showMsgDiv: boolean = false;
13. constructor() { }
14. ngOnInit() {
15. this.products = [
16. { "productId": "P101", "productName": "Lamborghini Gallardo Spyder", "categoryId": 1, "price": 18000000, "quantityAvailable": 10 },
17. { "productId": "P102", "productName": "Ben Sherman Mens Necktie Silk Tie", "categoryId": 2, "price": 1847, "quantityAvailable": 20 },
18. { "productId": "P103", "productName": "BMW Z4", "categoryId": 1, "price": 6890000, "quantityAvailable": 10 },
19. { "productId": "P104", "productName": "Samsung Galaxy S4", "categoryId": 3, "price": 38800, "quantityAvailable": 100 }
20. ]
21. if (this.products == null)
22. {
23. this.showMsgDiv = true;
24. }
25. }
26. }

**Step 15:**

In view-products.component.ts add the following lines of code from 27 - 31 to add category details in categories array declared in the previous step.

1. import { Component, OnInit } from '@angular/core';
2. import { IProduct } from '../quickKart-interfaces/product';
3. import { ICategory } from '../quickKart-interfaces/category';
4. @Component({
5. selector: 'app-view-products',
6. templateUrl: './view-products.component.html',
7. styleUrls: ['./view-products.component.css']
8. })
9. export class ViewProductsComponent implements OnInit {
10. products: IProduct[];
11. categories: ICategory[];
12. showMsgDiv: boolean = false;
13. constructor() { }
14. ngOnInit() {
15. this.products = [
16. { "productId": "P101", "productName": "Lamborghini Gallardo Spyder", "categoryId": 1, "price": 18000000, "quantityAvailable": 10 },
17. { "productId": "P102", "productName": "Ben Sherman Mens Necktie Silk Tie", "categoryId": 2, "price": 1847, "quantityAvailable": 20 },
18. { "productId": "P103", "productName": "BMW Z4", "categoryId": 1, "price": 6890000, "quantityAvailable": 10 },
19. { "productId": "P104", "productName": "Samsung Galaxy S4", "categoryId": 3, "price": 38800, "quantityAvailable": 100 }
20. ]
21. this.categories = [
22. { "categoryId": 1, "categoryName": "Motors" },
23. { "categoryId": 2, "categoryName": "Fashion" },
24. { "categoryId": 3, "categoryName": "Electronics" }
25. ]
26. if (this.products == null)
27. {
28. this.showMsgDiv = true;
29. }
30. }
31. }

**Step 16:**

Open view-products.component.html and add the following code from line 3 - 17 to add drop down list to display categories stored in categories array.

1. <div style="text-align:center;">
2. <h1>View Products</h1>
3. <br />
4. <div class="row">
5. <div class="col-md-10" style="text-align:right">
6. <label>Filter products:</label>
7. </div>
8. <div class="col-md-2">
9. <select class="form-control">
10. <option value="0">All Categories</option>
11. <option \*ngFor="let category of categories" value={{category.categoryId}}>
12. {{category.categoryName}}
13. </option>
14. </select>
15. </div>
16. </div>
17. <br />
18. <div class="table-responsive">
19. <table class="table" style="border:5px solid rgba(220,230,242,1);" \*ngIf="products">
20. <tr style="background-color:rgba(220,230,242,1); font-size:12pt">
21. <th style="text-align:center;">Product Id</th>
22. <th style="text-align:center;">Product Name</th>
23. <th style="text-align:center;">Category Id</th>
24. <th style="text-align:center;">Price</th>
25. <th style="text-align:center;">Quantity Available</th>
26. <th style="text-align:center;">Action</th>
27. </tr>
28. <tr \*ngFor="let product of products" style="background-color:white;text-align:center;font-size:12pt">
29. <td> {{product.productId }} </td>
30. <td> {{product.productName}} </td>
31. <td> {{product.categoryId}} </td>
32. <td> {{product.price}} </td>
33. <td> {{product.quantityAvailable}} </td>
34. <td><button title="Add to cart">Add to Cart</button> </td>
35. </tr>
36. </table>
37. </div>
38. <div \*ngIf="showMsgDiv" style="">
39. <h4 class="jumbotron" style="text-align:center;">No products available</h4>
40. </div>
41. </div>

Notice that the categories drop down displays category name with interpolation i.e. {{category.categoryName}}, but when the category is selected by the end user the value of categoryId will be captured.

**Step 17:**

In order to filter the products based upon on the category selected from the categories drop down, you need to add some functionality to ViewProductsComponent class.

Declare an array **filteredProducts**of type **IProduct**in line 14for storing the filtered products.

1. import { Component, OnInit } from '@angular/core';
2. import { IProduct } from '../quickKart-interfaces/product';
3. import { ICategory } from '../quickKart-interfaces/category';
4. @Component({
5. selector: 'app-view-products',
6. templateUrl: './view-products.component.html',
7. styleUrls: ['./view-products.component.css']
8. })
9. export class ViewProductsComponent implements OnInit {
10. products: IProduct[];
11. categories: ICategory[];
12. filteredProducts: IProduct[];
13. showMsgDiv: boolean = false;
14. constructor() { }
15. ngOnInit() {
16. this.products = [
17. { "productId": "P101", "productName": "Lamborghini Gallardo Spyder", "categoryId": 1, "price": 18000000, "quantityAvailable": 10 },
18. { "productId": "P102", "productName": "Ben Sherman Mens Necktie Silk Tie", "categoryId": 2, "price": 1847, "quantityAvailable": 20 },
19. { "productId": "P103", "productName": "BMW Z4", "categoryId": 1, "price": 6890000, "quantityAvailable": 10 },
20. { "productId": "P104", "productName": "Samsung Galaxy S4", "categoryId": 3, "price": 38800, "quantityAvailable": 100 }
21. ]
22. this.categories = [
23. { "categoryId": 1, "categoryName": "Motors" },
24. { "categoryId": 2, "categoryName": "Fashion" },
25. { "categoryId": 3, "categoryName": "Electronics" }
26. ]
27. if (this.products == null)
28. {
29. this.showMsgDiv = true;
30. }
31. }
32. }

When the ViewProductsComponent is loaded for the first time the filtering condition will not be applied, which means the filtered products is equal to products contained by product array.

Add the following line 40 in ngOninit() to the make filteredProducts array equal to products array.

1. import { Component, OnInit } from '@angular/core';
2. import { IProduct } from '../quickKart-interfaces/product';
3. import { ICategory } from '../quickKart-interfaces/category';
4. @Component({
5. selector: 'app-view-products',
6. templateUrl: './view-products.component.html',
7. styleUrls: ['./view-products.component.css']
8. })
9. export class ViewProductsComponent implements OnInit {
10. products: IProduct[];
11. categories: ICategory[];
12. filteredProducts: IProduct[];
13. showMsgDiv: boolean = false;
14. constructor() { }
15. ngOnInit() {
16. this.products = [
17. { "productId": "P101", "productName": "Lamborghini Gallardo Spyder", "categoryId": 1, "price": 18000000, "quantityAvailable": 10 },
18. { "productId": "P102", "productName": "Ben Sherman Mens Necktie Silk Tie", "categoryId": 2, "price": 1847, "quantityAvailable": 20 },
19. { "productId": "P103", "productName": "BMW Z4", "categoryId": 1, "price": 6890000, "quantityAvailable": 10 },
20. { "productId": "P104", "productName": "Samsung Galaxy S4", "categoryId": 3, "price": 38800, "quantityAvailable": 100 }
21. ]
23. this.categories = [
24. { "categoryId": 1, "categoryName": "Motors" },
25. { "categoryId": 2, "categoryName": "Fashion" },
26. { "categoryId": 3, "categoryName": "Electronics" }
27. ]
28. if (this.products == null)
29. {
30. this.showMsgDiv = true;
31. }
32. this.filteredProducts = this.products;
33. }
34. }

**Step 18:**

Add the code from line 42 - 52 in ViewProductsComponent class to add searchProductByCategory() method which accepts the category id as parameter and filter the products based upon selected category.

1. import { Component, OnInit } from '@angular/core';
2. import { IProduct } from '../quickKart-interfaces/product';
3. import { ICategory } from '../quickKart-interfaces/category';
4. @Component({
5. selector: 'app-view-products',
6. templateUrl: './view-products.component.html',
7. styleUrls: ['./view-products.component.css']
8. })
9. export class ViewProductsComponent implements OnInit {
10. products: IProduct[];
11. categories: ICategory[];
12. filteredProducts: IProduct[];
13. showMsgDiv: boolean = false;
14. constructor() { }
15. ngOnInit() {
16. this.products = [
17. { "productId": "P101", "productName": "Lamborghini Gallardo Spyder", "categoryId": 1, "price": 18000000, "quantityAvailable": 10 },
18. { "productId": "P102", "productName": "Ben Sherman Mens Necktie Silk Tie", "categoryId": 2, "price": 1847, "quantityAvailable": 20 },
19. { "productId": "P103", "productName": "BMW Z4", "categoryId": 1, "price": 6890000, "quantityAvailable": 10 },
20. { "productId": "P104", "productName": "Samsung Galaxy S4", "categoryId": 3, "price": 38800, "quantityAvailable": 100 }
21. ]
23. this.categories = [
24. { "categoryId": 1, "categoryName": "Motors" },
25. { "categoryId": 2, "categoryName": "Fashion" },
26. { "categoryId": 3, "categoryName": "Electronics" }
27. ]
28. if (this.products == null)
29. {
30. this.showMsgDiv = true;
31. }
32. this.filteredProducts = this.products;
33. }
34. searchProductByCategory(categoryId: string) {
35. this.filteredProducts = this.products;
36. if (categoryId == "0") {
37. this.filteredProducts = this.products;
38. }
39. else {
40. this.filteredProducts = this.filteredProducts.filter(prod => prod.categoryId.toString() == categoryId);
41. }
42. }
43. }

If you do not select any category then the categoryid will be 0. In this case all the products should be displayed. Otherwise the products should be filtered based on the selected category. Here filter function will filter only those products whose categoryid matches with the selected categoryid, which is passed using an arrow function.

**Step 19:**

Open view-products.component.html, modify the existing code by adding line 9 to invoke the searchProductByCategory() whenever the change in category dropdown is made.

1. <div style="text-align:center;">
2. <h1>View Products</h1>
3. <br />
4. <div class="row">
5. <div class="col-md-10" style="text-align:right">
6. <label>Filter products:</label>
7. </div>
8. <div class="col-md-2">
9. <select class="form-control" #categorydrop (change)="searchProductByCategory(categorydrop.value)">
10. <option value="0">All Categories</option>
11. <option \*ngFor="let category of categories" value={{category.categoryId}}>
12. {{category.categoryName}}
13. </option>
14. </select>
15. </div>
16. </div>
17. <br />
18. <div class="table-responsive">
19. <table class="table" style="border:5px solid rgba(220,230,242,1);" \*ngIf="products">
20. <tr style="background-color:rgba(220,230,242,1); font-size:12pt">
21. <th style="text-align:center;">Product Id</th>
22. <th style="text-align:center;">Product Name</th>
23. <th style="text-align:center;">Category Id</th>
24. <th style="text-align:center;">Price</th>
25. <th style="text-align:center;">Quantity Available</th>
26. <th style="text-align:center;">Action</th>
27. </tr>
28. <tr \*ngFor="let product of products" style="background-color:white;text-align:center;font-size:12pt">
29. <td> {{product.productId }} </td>
30. <td> {{product.productName}} </td>
31. <td> {{product.categoryId}} </td>
32. <td> {{product.price}} </td>
33. <td> {{product.quantityAvailable}} </td>
34. <td><button title="Add to cart">Add to Cart</button> </td>
35. </tr>
36. </table>
37. </div>
38. <div \*ngIf="showMsgDiv" style="">
39. <h4 class="jumbotron" style="text-align:center;">No products available</h4>
40. </div>
41. </div>

Observe **(change)="searchProductByCategory()"**. Here change is the event which is assigned with an event handler searchProductByCategory(). So whenever a change event is raised, searchProductByCategory() is invoked to filters the products.

Observe **#categorydrop**. # attached with a name serves as the id of the form control. This id can be used to retrieve the value and pass it to the event handler.

Observe **searchProductByCategory(categorydrop.value)**. categorydrop.value will hold the value selected by the user.

**Step 20:**

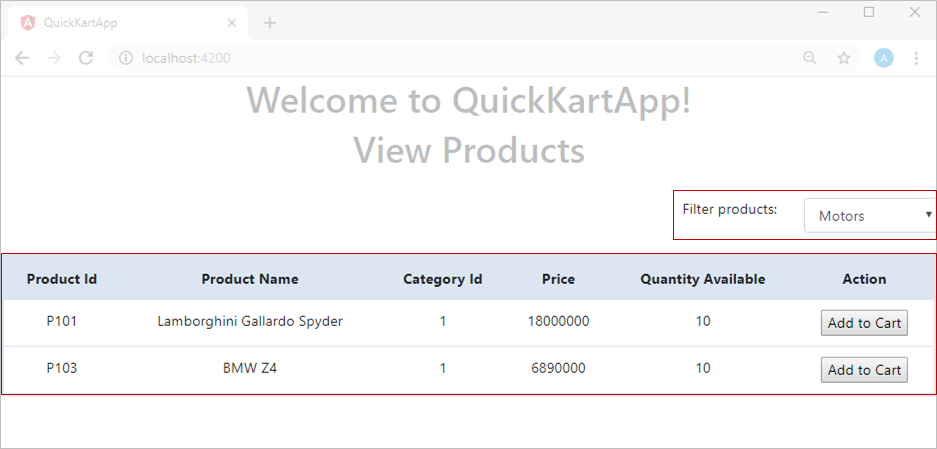
Open view-products.component.html and replace the products array in \*ngIf and \*ngFor template with filteredProducts array in line 19 and line 28 , as products should be displayed based upon filter condition i.e. filtered products should be displayed.

1. <div style="text-align:center;">
2. <h1>View Products</h1>
3. <br />
4. <div class="row">
5. <div class="col-md-10" style="text-align:right">
6. <label>Filter products:</label>
7. </div>
8. <div class="col-md-2">
9. <select class="form-control" #categorydrop (change)="searchProductByCategory(categorydrop.value)">
10. <option value="0">All Categories</option>
11. <option \*ngFor="let category of categories" value={{category.categoryId}}>
12. {{category.categoryName}}
13. </option>
14. </select>
15. </div>
16. </div>
17. <br />
18. <div class="table-responsive">
19. <table class="table" style="border:5px solid rgba(220,230,242,1);" \*ngIf="filteredProducts && filteredProducts.length">
20. <tr style="background-color:rgba(220,230,242,1); font-size:12pt">
21. <th style="text-align:center;">Product Id</th>
22. <th style="text-align:center;">Product Name</th>
23. <th style="text-align:center;">Category Id</th>
24. <th style="text-align:center;">Price</th>
25. <th style="text-align:center;">Quantity Available</th>
26. <th style="text-align:center;">Action</th>
27. </tr>
28. <tr \*ngFor="let product of filteredProducts" style="background-color:white;text-align:center;font-size:12pt">
29. <td> {{product.productId }} </td>
30. <td> {{product.productName}} </td>
31. <td> {{product.categoryId}} </td>
32. <td> {{product.price}} </td>
33. <td> {{product.quantityAvailable}} </td>
34. <td><button title="Add to cart">Add to Cart</button> </td>
35. </tr>
36. </table>
37. </div>
38. <div \*ngIf="showMsgDiv" style="">
39. <h4 class="jumbotron" style="text-align:center;">No products available</h4>
40. </div>
41. </div>

Observe that \*ngIf template is specified with the condition that if filteredProduct array is not null and its length is not equal to 0 then show the product details table.

**Step 21:**

Execute your application and try changing the categories and observe whether the products are filtered accordingly.



You can observe that products are filtered based on the selected category.

**Property binding:**

You will learn property binding by adding an image for add to Cart button.

**Step 22:**

Right click on the app folder and select Add --> New Folder. Rename this folder as **quickKart-images**.

Copy the images folder from the supplied files.

Copy **add-item.jpg** from images folder and paste it in **quickKart-images.**

**Step 23:**

Declare a variable **imageSrc** of type string for storing the image(add-item.jpg) path as in line 15. Assign the image path to **imageSrc**variable in ngOnInit() as shown below in line 40.

Modify the view-products.component.ts as shown below.

1. import { Component, OnInit } from '@angular/core';
2. import { IProduct } from '../quickKart-interfaces/product';
3. import { ICategory } from '../quickKart-interfaces/category';
4. @Component({
5. selector: 'app-view-products',
6. templateUrl: './view-products.component.html',
7. styleUrls: ['./view-products.component.css']
8. })
9. export class ViewProductsComponent implements OnInit {
10. products: IProduct[];
11. categories: ICategory[];
12. filteredProducts: IProduct[];
13. imageSrc: string;
14. showMsgDiv: boolean = false;
15. constructor() { }
16. ngOnInit() {
17. this.products = [
18. { "productId": "P101", "productName": "Lamborghini Gallardo Spyder", "categoryId": 1, "price": 18000000, "quantityAvailable": 10 },
19. { "productId": "P102", "productName": "Ben Sherman Mens Necktie Silk Tie", "categoryId": 2, "price": 1847, "quantityAvailable": 20 },
20. { "productId": "P103", "productName": "BMW Z4", "categoryId": 1, "price": 6890000, "quantityAvailable": 10 },
21. { "productId": "P104", "productName": "Samsung Galaxy S4", "categoryId": 3, "price": 38800, "quantityAvailable": 100 }
22. ]
23. this.categories = [
24. { "categoryId": 1, "categoryName": "Motors" },
25. { "categoryId": 2, "categoryName": "Fashion" },
26. { "categoryId": 3, "categoryName": "Electronics" }
27. ]
28. if (this.products == null) {
29. this.showMsgDiv = true;
30. }
31. this.filteredProducts = this.products;
32. this.imageSrc = "src/app/quickKart-images/add-item.jpg";
34. }
35. searchProductByCategory(categoryId: string) {
36. this.filteredProducts = this.products;
37. if (categoryId == "0") {
38. this.filteredProducts = this.products;
39. }
40. else {
41. this.filteredProducts = this.filteredProducts.filter(prod => prod.categoryId.toString() == categoryId);
42. }
43. }
44. }

**Step 24:**

In view-products.component.html change code written for **Add to Cart** button in table to show **add-item.jpg** image in the view in line 34.

1. <div style="text-align:center;">
2. <h1>View Products</h1>
3. <br />
4. <div class="row">
5. <div class="col-md-10" style="text-align:right">
6. <label>Filter products:</label>
7. </div>
8. <div class="col-md-2">
9. <select class="form-control" #categorydrop (change)="searchProductByCategory(categorydrop.value)">
10. <option value="0">All Categories</option>
11. <option \*ngFor="let category of categories" value={{category.categoryId}}>
12. {{category.categoryName}}
13. </option>
14. </select>
15. </div>
16. </div>
17. <br />
18. <div class="table-responsive">
19. <table class="table" style="border:5px solid rgba(220,230,242,1);" \*ngIf="filteredProducts && filteredProducts.length">
20. <tr style="background-color:rgba(220,230,242,1); font-size:12pt">
21. <th style="text-align:center;">Product Id</th>
22. <th style="text-align:center;">Product Name</th>
23. <th style="text-align:center;">Category Id</th>
24. <th style="text-align:center;">Price</th>
25. <th style="text-align:center;">Quantity Available</th>
26. <th style="text-align:center;">Action</th>
27. </tr>
28. <tr \*ngFor="let product of filteredProducts" style="background-color:white;text-align:center;font-size:12pt">
29. <td> {{product.productId }} </td>
30. <td> {{product.productName}} </td>
31. <td> {{product.categoryId}} </td>
32. <td> {{product.price}} </td>
33. <td> {{product.quantityAvailable}} </td>
34. <td><img title="Add to cart" [src]="imageSrc"></td>
35. </tr>
36. </table>
37. </div>
38. <div \*ngIf="showMsgDiv" style="">
39. <h4 class="jumbotron" style="text-align:center;">No products available</h4>
40. </div>
41. </div>

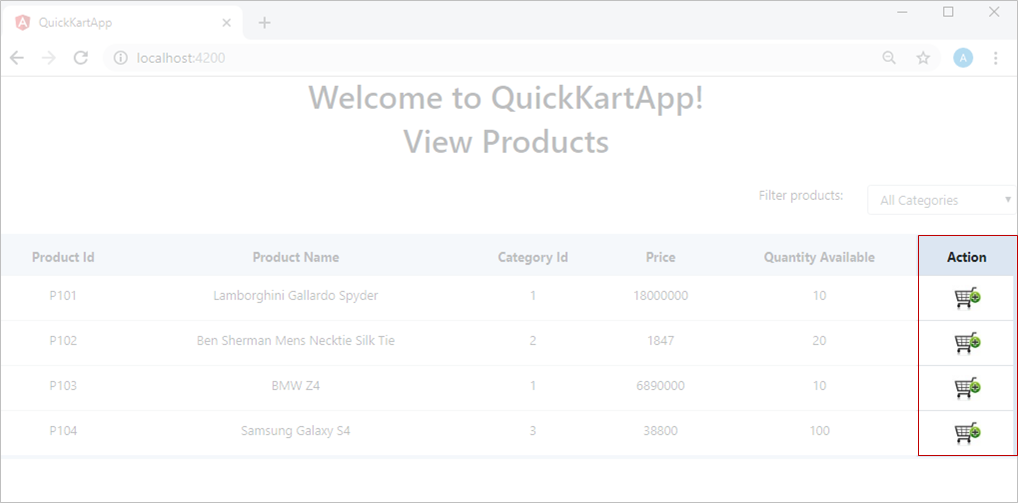
Note that the **src** property of img tag is wrapped in [], e.g. [src] and assigned a property **imageSrc** declared in ViewProductsComponent class.

This type of binding is called **property binding**, in which the property of an element is binded to a variable declared in the component.

Please note here src is a property not an attribute. An attribute is one time initialization and can't change once initialized but property can change. Therefore src property gets changed whenever the value of imageSrc changes.

**Step 25:**

Now execute your application and check if the page is loaded as shown below.



Observe that the **Add to Cart** button is replaced by the image.

Note: If the image doesnot load for you, keep the image in assets folder and try with the corresponding path.