

File Edit Selection View Go Debug Terminal Help

welcome.component.html - app - Copy - Visual Studio Code

EXPLORER

OPEN EDITORS

APP - COPY

- home
- welcome.component.html
- welcome.component.ts
- products
- product-data.ts
- # product-detail.component.css
- product-detail.component.html
- product-detail.component.ts
- product-edit.component.html
- product-edit.component.ts
- product-edit.guard.ts
- # product-list.component.css
- product-list.component.html
- product-list.component.ts
- product.module.ts
- product.service.ts
- product.ts
- shared
- generic-validator.ts
- number.validator.ts
- shared.module.ts
- # star.component.css
- star.component.html
- star.component.ts
- app.component.ts
- app.module.ts

OUTLINE

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

```
<div class="card">
  <div class="card-header">
    {{pageTitle}}
  </div>
  <div class="card-body">
    <div class="container-fluid">
      <div class="text-center">
        
      </div>

      <div class="text-center">Developed by:</div>
      <div class="text-center">
        <h3>Deborah Kurata</h3>
      </div>

      <div class="text-center">@deborahkurata</div>
      <div class="text-center">
        <a href="http://www.bit.ly/DeborahKsBlog">www.bit.ly/DeborahKsBlog</a>
      </div>
    </div>
  </div>
</div>
```

0 ▲ 0 { } : 61 Go Live Ln 1, Col 1 Spaces: 2 UTF-8

Populating the Form with Data

Sign Up!

First Name	First Name (required)
Last Name	La Edit Product: Hammer
Email	Er
<input checked="" type="checkbox"/> Send me your catalog	
Address Type	St
Street Address 1	St
Street Address 2	St
City, State, Zip Code	Cl
	Add Tag
Description	Curved claw steel hammer
	<input type="button" value="Save"/> <input type="button" value="Cancel"/> <input type="button" value="Delete"/>

Product List

Filter by:

Show Image	Product	Code	Available	Price	5 Star Rating	Edit
	Leaf Rake	GDN-0011	March 19, 2018	\$19.95	★★★	<input type="button" value="Edit"/>
	Garden Cart	GDN-0023	March 18, 2018	\$32.99	★★★★	<input type="button" value="Edit"/>
	Hammer	TBX-0048	May 21, 2018	\$8.90	★★★★★	<input type="button" value="Edit"/>
	Saw	TBX-0022	May 15, 2018	\$11.55	★★★★	<input type="button" value="Edit"/>
	Video Game Controller	GMG-0042	October 15, 2018	\$35.95	★★★★★	<input type="button" value="Edit"/>

Saving Edits

Edit Product: Hammer

Product Name	Hammer
Product Code	TBX-0048
Star Rating (1-5)	4.8
Tag	tools
Tag	hammer
Tag	construction
Add Tag	
Description	Curved claw steel hammer

[Save](#) [Cancel](#) [Delete](#)

```
{"productName": "Hammer", "productCode": "TBX-0048", "starRating": 4.8, "description": "Curved claw steel hammer", "tags": [ "tools", "hammer", "construction" ]}
```

Creating New Items

Acme Product Management [Home](#) [Product List](#) [Add Product](#)

Add Product

Product Name	Name (required)
Product Code	Code (required)
Star Rating (1-5)	Rating (1-5)
Tag	Tag
Add Tag	
Description	Description

[Save](#) [Cancel](#) [Delete](#)

Deleting an Existing Item

Edit Product: Hammer

Product Name	Hammer
Product Code	TBX-0048
Star Rating (1-5)	4.8
Tag	tools
Tag	hammer
Tag	construction
Add Tag	
Description	Curved claw steel hammer

Product List

Show Image	Product	Code	Available	Price	5 Star Rating	Edit
	Leaf Rake	GDN-0011	March 19, 2018	\$19.95	★★★☆	Edit
	Garden Cart	GDN-0023	March 18, 2018	\$32.99	★★★★☆	Edit
	Hammer	TBX-0048	May 21, 2018	\$8.90	★★★★★	Edit
	Saw	TBX-0022	May 15, 2018	\$11.55	★★★★☆	Edit
	Video Game Controller	GMG-0042	October 15, 2018	\$35.95	★★★★★	Edit

Save **Cancel** **Delete**

```
1 import { BrowserModule } from '@angular/platform-browser';
2 import { NgModule } from '@angular/core';
3 import { HttpClientModule } from '@angular/common/http';
4 import { RouterModule } from '@angular/router';
5
6 import { AppComponent } from './app.component';
7 import { WelcomeComponent } from './home/welcome.component';
8 import { ProductModule } from './products/product.module';
9
10 @NgModule({
11   declarations: [
12     AppComponent,
13     WelcomeComponent
14   ],
15   imports: [
16     BrowserModule,
17     HttpClientModule,
18     RouterModule.forRoot([
19       { path: 'welcome', component: WelcomeComponent },
20       { path: '', redirectTo: 'welcome', pathMatch: 'full' },
21       { path: '**', redirectTo: 'welcome', pathMatch: 'full' }
22     ]),
23     ProductModule
24   ],
25   bootstrap: [AppComponent]
26 })
27 export class AppModule { }
```

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'pm-root',
5   template: `
6     <nav class='navbar navbar-expand navbar-light bg-light'>
7       <a class='navbar-brand'>{{pageTitle}}</a>
8       <ul class='navbar-nav'>
9         <li class='nav-item'><a class='nav-link' routerLinkActive='active'
10           [routerLink]="/welcome">Home</a>
11         </li>
12         <li class='nav-item'><a class='nav-link' routerLinkActive='active'
13           [routerLinkActiveOptions]="{exact: true}"
14             [routerLink]="/products">Product List</a>
15           </li>
16           <li class='nav-item'><a class='nav-link' routerLinkActive='active'
17             [routerLinkActiveOptions]="{exact: true}"
18               [routerLink]="/products/0/edit">Add Product</a>
19             </li>
20           </ul>
21         </nav>
22         <div class='container'>
23           <router-outlet></router-outlet>
24         </div>
25       `,
26       styleUrls: ['./app.component.css']
27     })
28   export class AppComponent {
29     pageTitle = 'Acme Product Management';
30   }
31 }
```

```
1 import { Component, OnChanges, Input, EventEmitter, Output } from '@angular/core';
2
3 @Component({
4   selector: 'pm-star',
5   templateUrl: './star.component.html',
6   styleUrls: ['./star.component.css']
7 })
8 export class StarComponent implements OnChanges {
9   @Input() rating = 0;
10  starWidth = 0;
11  @Output() ratingClicked: EventEmitter<string> =
12    new EventEmitter<string>();
13
14  ngOnChanges(): void {
15    this.starWidth = this.rating * 75 / 5;
16  }
17
18  onClick(): void {
19    this.ratingClicked.emit(`The rating ${this.rating} was clicked!`);
20  }
21}
22
```

```
1 <div class="crop"
2   [style.width.px]="starWidth"
3   [title]="rating"
4   (click)="onClick()">
5   <div style="width: 75px">
6     <span class="fa fa-star"></span>
7     <span class="fa fa-star"></span>
8     <span class="fa fa-star"></span>
9     <span class="fa fa-star"></span>
10    <span class="fa fa-star"></span>
11  </div>
12 </div>
```

```
1 import { NgModule } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { FormsModule } from '@angular/forms';
4
5 import { StarComponent } from './star.component';
6
7 @NgModule({
8   imports: [
9     CommonModule
10  ],
11   declarations: [
12     StarComponent
13  ],
14   exports: [
15     StarComponent,
16     CommonModule,
17     FormsModule
18  ]
19 })
20 export class SharedModule { }
21
```

```
1 import { AbstractControl, ValidatorFn } from '@angular/forms';
2
3 export class NumberValidators {
4
5   static range(min: number, max: number): ValidatorFn {
6     return (c: AbstractControl): { [key: string]: boolean } | null => {
7       if (c.value && (isNaN(c.value) || c.value < min || c.value > max)) {
8         return { 'range': true };
9       }
10      return null;
11    };
12  }
13}
14
```

```

1 import { FormGroup } from '@angular/forms';
2 // Generic validator for Reactive forms
3 // Implemented as a class, not a service, so it can retain state for multiple forms.
4 export class GenericValidator {
5   // Provide the set of valid validation messages // Structure:
6   // controlName1: {
7     //   validationRuleName1: 'Validation Message.',
8     //   validationRuleName2: 'Validation Message.'
9   },
10  // controlName2: {
11    //   validationRuleName1: 'Validation Message.',
12    //   validationRuleName2: 'Validation Message.'
13  }
14  constructor(private validationMessages: { [key: string]: { [key: string]: string } })
15  ) {}
16  // Processes each control within a FormGroup, And returns a set of validation
17  // messages to display// Structure
18  // controlName1: 'Validation Message.',
19  // controlName2: 'Validation Message.'
20  processMessages(container: FormGroup): { [key: string]: string } {
21    const messages = {};
22    for (const controlKey in container.controls) {
23      if (container.controls.hasOwnProperty(controlKey)) {
24        const c = container.controls[controlKey];
25        // If it is a FormGroup, process its child controls.
26        if (c instanceof FormGroup) {
27          const childMessages = this.processMessages(c);
28          Object.assign(messages, childMessages);
29        } else {
30          // Only validate if there are validation messages for the control
31          if (this.validationMessages[controlKey]) {
32            messages[controlKey] = '';
33            if ((c.dirty || c.touched) && c.errors) {
34              Object.keys(c.errors).map(messageKey => {
35                if (this.validationMessages[controlKey][messageKey]) {
36                  messages[controlKey] += this.validationMessages[controlKey]
37                  [messageKey] + ' ';
38                }
39              });
40            }
41          }
42        }
43      }
44      return messages;
45    }
46    getErrorCount(container: FormGroup): number {
47      let errorCount = 0;
48      for (const controlKey in container.controls) {
49        if (container.controls.hasOwnProperty(controlKey)) {
50          if (container.controls[controlKey].errors) {
51            errorCount += Object.keys(container.controls[controlKey].errors).length;
52            console.log(errorCount);
53          }
54        }
55      }
56    }
57  }

```

```
1 /* Defines the product entity */
2 export interface Product {
3   id: number;
4   productName: string;
5   productCode: string;
6   tags?: string[];
7   releaseDate: string;
8   price: number;
9   description: string;
10  starRating: number;
11  imageUrl: string;
12 }
13
14
```

```
1 import { Injectable } from '@angular/core';
2 import { HttpClient, HttpHeaders } from '@angular/common/http';
3
4 import { Observable, of, throwError } from 'rxjs';
5 import { catchError, tap, map } from 'rxjs/operators';
6
7 import { Product } from './product';
8
9 @Injectable({
10   providedIn: 'root'
11 })
12 export class ProductService {
13   private productsUrl = 'api/products';
14
15   constructor(private http: HttpClient) { }
16
17   getProducts(): Observable<Product[]> {
18     return this.http.get<Product[]>(this.productsUrl)
19       .pipe(
20         tap(data => console.log(JSON.stringify(data))),
21         catchError(this.handleError)
22       );
23   }
24
25   getProduct(id: number): Observable<Product> {
26     if (id === 0) {
27       return of(this.initializeProduct());
28     }
29     const url = `${this.productsUrl}/${id}`;
30     return this.http.get<Product>(url)
31       .pipe(
32         tap(data => console.log('getProduct: ' + JSON.stringify(data))),
33         catchError(this.handleError)
34       );
35   }
36
37   createProduct(product: Product): Observable<Product> {
38     const headers = new HttpHeaders({ 'Content-Type': 'application/json' });
39     product.id = null;
40     return this.http.post<Product>(this.productsUrl, product, { headers: headers })
41       .pipe(
42         tap(data => console.log('createProduct: ' + JSON.stringify(data))),
43         catchError(this.handleError)
44       );
45   }
46
47   deleteProduct(id: number): Observable<{}> {
48     const headers = new HttpHeaders({ 'Content-Type': 'application/json' });
49     const url = `${this.productsUrl}/${id}`;
50     return this.http.delete<Product>(url, { headers: headers })
51       .pipe(
52         tap(data => console.log('deleteProduct: ' + id)),
53         catchError(this.handleError)
54       );
55   }
56
57   updateProduct(product: Product): Observable<Product> {
58     const headers = new HttpHeaders({ 'Content-Type': 'application/json' });
59     const url = `${this.productsUrl}/${product.id}`;
60     return this.http.put<Product>(url, product, { headers: headers })
```

```
61     .pipe(
62       tap(() => console.log('updateProduct: ' + product.id)),
63       // Return the product on an update
64       map(() => product),
65       catchError(this.handleError)
66     );
67   }
68
69   private handleError(err) {
70     // in a real world app, we may send the server to some remote logging
71     // infrastructure
72     // instead of just logging it to the console
73     let errorMessage: string;
74     if (err.error instanceof ErrorEvent) {
75       // A client-side or network error occurred. Handle it accordingly.
76       errorMessage = `An error occurred: ${err.error.message}`;
77     } else {
78       // The backend returned an unsuccessful response code.
79       // The response body may contain clues as to what went wrong,
80       errorMessage = `Backend returned code ${err.status}: ${err.body.error}`;
81     }
82     console.error(err);
83     return throwError(errorMessage);
84   }
85
86   private initializeProduct(): Product {
87     // Return an initialized object
88     return {
89       id: 0,
90       productName: null,
91       productCode: null,
92       tags: [''],
93       releaseDate: null,
94       price: null,
95       description: null,
96       starRating: null,
97       imageUrl: null
98     };
99   }
100 }
```

```
1 import { NgModule } from '@angular/core';
2 import { RouterModule } from '@angular/router';
3 import { ReactiveFormsModule } from '@angular/forms';
4
5 import { SharedModule } from '../shared/shared.module';
6
7 // Imports for loading & configuring the in-memory web api
8 import { InMemoryWebApiModule } from 'angular-in-memory-web-api';
9 import { ProductData } from './product-data';
10
11 import { ProductListComponent } from './product-list.component';
12 import { ProductDetailComponent } from './product-detail.component';
13 import { ProductEditComponent } from './product-edit.component';
14 import { ProductEditGuard } from './product-edit.guard';
15
16 @NgModule({
17   imports: [
18     SharedModule,
19     ReactiveFormsModule,
20     InMemoryWebApiModule.forRoot(ProductData),
21     RouterModule.forChild([
22       { path: 'products', component: ProductListComponent },
23       { path: 'products/:id', component: ProductDetailComponent },
24       {
25         path: 'products/:id/edit',
26         canActivate: [ProductEditGuard],
27         component: ProductEditComponent
28       }
29     ])
30   ],
31   declarations: [
32     ProductListComponent,
33     ProductDetailComponent,
34     ProductEditComponent
35   ]
36 })
37 export class ProductModule { }
```

```
1 import { Component, OnInit } from '@angular/core';
2
3 import { Product } from './product';
4 import { ProductService } from './product.service';
5
6 @Component({
7   templateUrl: './product-list.component.html',
8   styleUrls: ['./product-list.component.css']
9 })
10 export class ProductListComponent implements OnInit {
11   pageTitle = 'Product List';
12   imageWidth = 50;
13   imageMargin = 2;
14   showImage = false;
15   errorMessage = '';
16
17   _listFilter = '';
18   get listFilter(): string {
19     return this._listFilter;
20   }
21   set listFilter(value: string) {
22     this._listFilter = value;
23     this.filteredProducts = this.listFilter ? this.performFilter(this.listFilter) :
24       this.products;
25   }
26
27   filteredProducts: Product[] = [];
28   products: Product[] = [];
29
30   constructor(private productService: ProductService) { }
31
32   performFilter(filterBy: string): Product[] {
33     filterBy = filterBy.toLocaleLowerCase();
34     return this.products.filter((product: Product) =>
35       product.productName.toLocaleLowerCase().indexOf(filterBy) !== -1);
36   }
37
38   toggleImage(): void {
39     this.showImage = !this.showImage;
40   }
41
42   ngOnInit(): void {
43     this.productService.getProducts().subscribe(
44       products => {
45         this.products = products;
46         this.filteredProducts = this.products;
47       },
48       error => this.errorMessage = <any>error
49     );
50   }
51 }
52 }
```

```
1 <div class="card">
2   <div class="card-header">
3     {{pageTitle}}
4   </div>
5
6   <div class="card-body">
7     <div class="row">
8       <div class="col-md-2">Filter by:</div>
9       <div class="col-md-4">
10      <input type="text"
11        [(ngModel)]="listFilter" />
12      </div>
13    </div>
14    <div class="row"
15      *ngIf="listFilter">
16      <div class="col-md-6">
17        <h4>Filtered by: {{listFilter}}</h4>
18      </div>
19    </div>
20
21   <div class="table-responsive">
22     <table class="table mb-0"
23       *ngIf="products && products.length">
24       <thead>
25         <tr>
26           <th>
27             <button class="btn btn-outline-primary btn-sm"
28               (click)="toggleImage()">
29               {{showImage ? "Hide" : "Show"}} Image
30             </button>
31           </th>
32           <th>Product</th>
33           <th>Code</th>
34           <th>Available</th>
35           <th>Price</th>
36           <th>5 Star Rating</th>
37           <th></th>
38         </tr>
39       </thead>
40       <tbody>
41         <tr *ngFor="let product of filteredProducts">
42           <td>
43             <img *ngIf="showImage && product.imageUrl"
44               [src]="product.imageUrl"
45               [title]="product.productName"
46               [style.width.px]="imageWidth"
47               [style.margin.px]="imageMargin">
48           </td>
49           <td>
50             <a [routerLink]=["/products", product.id]>
51               {{ product.productName }}</a>
52           </td>
53           <td>{{ product.productCode }}</td>
54           <td>{{ product.releaseDate }}</td>
55           <td>{{ product.price | currency:"USD":symbol:"1.2-2" }}</td>
56           <td>
57             <pm-star [rating]="product.starRating">
58             </pm-star>
59           </td>
60         </tr>
61       </tbody>
62     </table>
63   </div>
64
65   <div class="row">
66     <div class="col-md-6">
67       <h4>Total Products: {{products.length}}</h4>
68     </div>
69   </div>
70
71   <div class="row">
72     <div class="col-md-6">
73       <button class="btn btn-primary" (click)="onSubmit()>
74         Search</button>
75     </div>
76   </div>
77
78   <div class="row">
79     <div class="col-md-6">
80       <table class="table">
81         <thead>
82           <tr>
83             <th>Product</th>
84             <th>Code</th>
85             <th>Available</th>
86             <th>Price</th>
87             <th>5 Star Rating</th>
88           </tr>
89         </thead>
90         <tbody>
91           <tr *ngFor="let product of products">
92             <td>{{ product.productName }}</td>
93             <td>{{ product.productCode }}</td>
94             <td>{{ product.releaseDate }}</td>
95             <td>{{ product.price | currency:"USD":symbol:"1.2-2" }}</td>
96             <td>{{ product.starRating }}</td>
97           </tr>
98         </tbody>
99       </table>
100      <div>
101        <button class="btn btn-primary" (click)="onSort('productCode')>
102          Sort by Product Code</button>
103        <button class="btn btn-primary" (click)="onSort('releaseDate')>
104          Sort by Release Date</button>
105        <button class="btn btn-primary" (click)="onSort('price')>
106          Sort by Price</button>
107        <button class="btn btn-primary" (click)="onSort('starRating')>
108          Sort by Star Rating</button>
109      </div>
110    </div>
111  </div>
112</div>
```

```
61      <td>
62          <button class="btn btn-outline-primary btn-sm"
63             [routerLink]=["/products", product.id, 'edit']>
64              Edit
65          </button>
66      </td>
67  </tr>
68 </tbody>
69 </table>
70 </div>
71
72 </div>
73 </div>
74
75 <div *ngIf="errorMessage"
76   class="alert alert-danger">
77   Error: {{ errorMessage }}
78 </div>
```

```
1 import { Injectable } from '@angular/core';
2 import { CanDeactivate } from '@angular/router';
3 import { Observable } from 'rxjs';
4
5 import { ProductEditComponent } from './product-edit.component';
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class ProductEditGuard implements CanDeactivate<ProductEditComponent> {
11   canDeactivate(component: ProductEditComponent): Observable<boolean> | Promise<boolean> | boolean {
12     if (component.productForm.dirty) {
13       const productName = component.productForm.get('productName').value || 'New
Product';
14       return confirm(`Navigate away and lose all changes to ${productName}?`);
15     }
16     return true;
17   }
18 }
19 }
```

```
1 import { Component, OnInit, AfterViewInit, OnDestroy, ViewChildren, ElementRef } from
2   '@angular/core';
3 import { FormBuilder, FormGroup, FormControl, FormArray, Validators, FormControlName
4 } from '@angular/forms';
5 import { ActivatedRoute, Router } from '@angular/router';
6
7 import { Observable, Subscription, fromEvent, merge } from 'rxjs';
8 import { debounceTime } from 'rxjs/operators';
9
10
11 import { Product } from './product';
12 import { ProductService } from './product.service';
13
14 @Component({
15   templateUrl: './product-edit.component.html'
16 })
17 export class ProductEditComponent implements OnInit, AfterViewInit, OnDestroy {
18   @ViewChildren(FormControlName, { read: ElementRef }) formInputElements:
19   ElementRef[];
20
21   pageTitle = 'Product Edit';
22   errorMessage: string;
23   productForm: FormGroup;
24
25   product: Product;
26   private sub: Subscription;
27
28   // Use with the generic validation message class
29   displayMessage: { [key: string]: string } = {};
30   private validationMessages: { [key: string]: { [key: string]: string } } = {};
31   private genericValidator: GenericValidator;
32
33   get tags(): FormArray {
34     return <FormArray>this.productForm.get('tags');
35   }
36
37   constructor(private fb: FormBuilder,
38     private route: ActivatedRoute,
39     private router: Router,
40     private productService: ProductService) {
41
42     // Defines all of the validation messages for the form.
43     // These could instead be retrieved from a file or database.
44     this.validationMessages = {
45       productName: {
46         required: 'Product name is required.',
47         minlength: 'Product name must be at least three characters.',
48         maxlength: 'Product name cannot exceed 50 characters.'
49       },
50       productCode: {
51         required: 'Product code is required.'
52       },
53       starRating: {
54         range: 'Rate the product between 1 (lowest) and 5 (highest).'
55       }
56     };
57
58     // Define an instance of the validator for use with this form,
59   }
60
61   ngOnInit() {
62     this.route.params.subscribe(params => {
63       const id = params['id'];
64       if (id) {
65         this.productService.getProduct(id).subscribe(product => {
66           this.product = product;
67           this.productForm.patchValue(this.product);
68         });
69       }
70     });
71   }
72
73   ngAfterViewInit() {
74     this.formInputElements.forEach(elementRef => {
75       const controlName = elementRef.nativeElement.getAttribute('name');
76       const control = this.productForm.get(controlName);
77       if (control instanceof FormControl) {
78         control.valueChanges
79           .pipe(debounceTime(500))
80           .subscribe(value => {
81             this.genericValidator.validateControl(control);
82           });
83       }
84     });
85   }
86
87   ngOnDestroy() {
88     this.sub.unsubscribe();
89   }
90 }
```

```
58 // passing in this form's set of validation messages.
59 this.genericValidator = new GenericValidator(this.validationMessages);
60 }
61
62 ngOnInit(): void {
63   this.productForm = this.fb.group({
64     productName: ['', [Validators.required,
65       Validators.minLength(3),
66       Validators.maxLength(50)]],
67     productCode: ['', Validators.required],
68     starRating: ['', NumberValidators.range(1, 5)],
69     tags: this.fb.array([]),
70     description: ''
71   });
72
73 // Read the product Id from the route parameter
74 this.sub = this.route.paramMap.subscribe(
75   params => {
76     const id = +params.get('id');
77     this.getProduct(id);
78   }
79 );
80 }
81
82 ngOnDestroy(): void {
83   this.sub.unsubscribe();
84 }
85
86 ngAfterViewInit(): void {
87   // Watch for the blur event from any input element on the form.
88   const controlBlurs: Observable<any>[] = this.formInputElements
89     .map((formControl: ElementRef) => fromEvent(formControl.nativeElement,
90 'blur'));
91
92   // Merge the blur event observable with the valueChanges observable
93   merge(this.productForm.valueChanges, ...controlBlurs).pipe(
94     debounceTime(800)
95   ).subscribe(value => {
96     this.displayMessage = this.genericValidator.processMessages(this.productForm);
97   });
98 }
99
100 addTag(): void {
101   this.tags.push(new FormControl());
102 }
103
104 deleteTag(index: number): void {
105   this.tags.removeAt(index);
106   this.tags.markAsDirty();
107 }
108
109 getProduct(id: number): void {
110   this.productService.getProduct(id)
111     .subscribe(
112       (product: Product) => this.displayProduct(product),
113       (error: any) => this.errorMessage = <any>error
114     );
115 }
116
117 displayProduct(product: Product): void {
```

```
117 if (this.productForm) {
118   this.productForm.reset();
119 }
120 this.product = product;
121
122 if (this.product.id === 0) {
123   this.pageTitle = 'Add Product';
124 } else {
125   this.pageTitle = `Edit Product: ${this.product.productName}`;
126 }
127
128 // Update the data on the form
129 this.productForm.patchValue({
130   productName: this.product.productName,
131   productCode: this.product.productCode,
132   starRating: this.product.starRating,
133   description: this.product.description
134 });
135 this.productForm.setControl('tags', this.fb.array(this.product.tags || []));
136 }
137
138 deleteProduct(): void {
139   if (this.product.id === 0) {
140     // Don't delete, it was never saved.
141     this.onSaveComplete();
142   } else {
143     if (confirm(`Really delete the product: ${this.product.productName}?`)) {
144       this.productService.deleteProduct(this.product.id)
145         .subscribe(
146           () => this.onSaveComplete(),
147           (error: any) => this.errorMessage = <any>error
148         );
149     }
150   }
151 }
152
153 saveProduct(): void {
154   if (this.productForm.valid) {
155     if (this.productForm.dirty) {
156       const p = { ...this.product, ...this.productForm.value };
157
158       if (p.id === 0) {
159         this.productService.createProduct(p)
160           .subscribe(
161             () => this.onSaveComplete(),
162             (error: any) => this.errorMessage = <any>error
163           );
164       } else {
165         this.productService.updateProduct(p)
166           .subscribe(
167             () => this.onSaveComplete(),
168             (error: any) => this.errorMessage = <any>error
169           );
170       }
171     } else {
172       this.onSaveComplete();
173     }
174   } else {
175     this.errorMessage = 'Please correct the validation errors.';
176   }
}
```

```
177 }
178
179 onSaveComplete(): void {
180     // Reset the form to clear the flags
181     this.productForm.reset();
182     this.router.navigate(['/products']);
183 }
184 }
185
```

```
1 <div class="card">
2   <div class="card-header">
3     {{pageTitle}}
4   </div>
5
6   <div class="card-body">
7     <form novalidate
8       (ngSubmit)="saveProduct()"
9       [formGroup]="productForm">
10
11    <div class="form-group row mb-2">
12      <label class="col-md-2 col-form-label"
13        for="productNameId">Product Name</label>
14      <div class="col-md-8">
15        <input class="form-control"
16          id="productNameId"
17          type="text"
18          placeholder="Name (required)"
19          formControlName="productName"
20          [ngClass]="{{'is-invalid': displayMessage.productName }}" />
21        <span class="invalid-feedback">
22          {{displayMessage.productName}}
23        </span>
24      </div>
25    </div>
26
27    <div class="form-group row mb-2">
28      <label class="col-md-2 col-form-label"
29        for="productCodeId">Product Code</label>
30      <div class="col-md-8">
31        <input class="form-control"
32          id="productCodeId"
33          type="text"
34          placeholder="Code (required)"
35          formControlName="productCode"
36          [ngClass]="{{'is-invalid': displayMessage.productCode }}" />
37        <span class="invalid-feedback">
38          {{displayMessage.productCode}}
39        </span>
40      </div>
41    </div>
42
43    <div class="form-group row mb-2">
44      <label class="col-md-2 col-form-label"
45        for="starRatingId">Star Rating (1-5)</label>
46
47      <div class="col-md-8">
48        <input class="form-control"
49          id="starRatingId"
50          type="text"
51          placeholder="Rating (1-5)"
52          formControlName="starRating"
53          [ngClass]="{{'is-invalid': displayMessage.starRating }}" />
54        <span class="invalid-feedback">
55          {{displayMessage.starRating}}
56        </span>
57      </div>
58    </div>
59
60    <div formArrayName="tags">
```

```
61 <div class="form-group row mb-2">
62   *ngFor="let tag of tags.controls; let i=index">
63   <label class="col-md-2 col-form-label">
64     [attr.for]="i">Tag</label>
65
66   <div class="col-md-8">
67     <input class="form-control"
68       [id]="i"
69       type="text"
70       placeholder="Tag"
71       [FormControlName]="i" />
72   </div>
73   <button class="col-md-1 btn btn-outline-warning btn-sm"
74     type="button"
75     title="Delete this search tag"
76     (click)="deleteTag(i)">Delete Tag
77   </button>
78 </div>
79 </div>
80
81 <div class="row">
82   <div class="col-md-1 mb-2">
83     <button class="btn btn-outline-secondary btn-sm"
84       type="button"
85       title="Add a search tag"
86       (click)="addTag()">Add Tag
87     </button>
88   </div>
89 </div>
90
91 <div class="form-group row mb-2">
92   <label class="col-md-2 col-form-label"
93     for="descriptionId">Description</label>
94
95   <div class="col-md-8">
96     <textarea class="form-control"
97       id="descriptionId"
98       placeholder="Description"
99       rows=3
100      formControlName="description"
101      [ngClass]="{{ 'is-invalid': displayMessage.description }}">
102   </textarea>
103   <span class="invalid-feedback">
104     {{ displayMessage.description }}>
105   </span>
106 </div>
107
108 <div class="form-group row mb-2">
109   <div class="offset-md-2 col-md-4">
110     <button class="btn btn-primary mr-3"
111       style="width:80px;"'
112       type="submit"
113       [title]="productForm.valid ? 'Save your entered data' : 'Disabled
until the form data is valid'"'
114       [disabled]!="productForm.valid">
115       Save
116     </button>
117     <button class="btn btn-outline-secondary mr-3"
118       style="width:80px;"'
```

```
119          type="button"
120          title="Cancel your edits"
121          [routerLink]=["/products"]>
122          Cancel
123      </button>
124      <button class="btn btn-outline-warning"
125             style="width:80px"
126             type="button"
127             title="Delete this product"
128             (click)="deleteProduct()">
129          Delete
130      </button>
131    </div>
132  </div>
133 </form>
134 </div>
135
136 <div class="alert alert-danger"
137   *ngIf="errorMessage">{{errorMessage}}
138 </div>
139 </div>
```

```
1 import { Component, OnInit } from '@angular/core';
2 import { ActivatedRoute, Router } from '@angular/router';
3
4 import { Product } from './product';
5 import { ProductService } from './product.service';
6
7 @Component({
8   templateUrl: './product-detail.component.html',
9   styleUrls: ['./product-detail.component.css']
10 })
11 export class ProductDetailComponent implements OnInit {
12   pageTitle = 'Product Detail';
13   errorMessage = '';
14   product: Product | undefined;
15
16   constructor(private route: ActivatedRoute,
17     private router: Router,
18     private productService: ProductService) {
19   }
20
21   ngOnInit() {
22     const param = this.route.snapshot.paramMap.get('id');
23     if (param) {
24       const id = +param;
25       this.getProduct(id);
26     }
27   }
28
29   getProduct(id: number) {
30     this.productService.getProduct(id).subscribe(
31       product => this.product = product,
32       error => this.errorMessage = <any>error);
33   }
34
35   onBack(): void {
36     this.router.navigate(['/products']);
37   }
38
39 }
40 }
```

```
1 <div class="card">
2   <div class="card-header"
3     *ngIf="product">
4     {{pageTitle + ": " + product.productName}}
5   </div>
6
7   <div class="card-body"
8     *ngIf="product">
9
10    <div class="row">
11
12      <div class="col-md-8">
13        <div class="row">
14          <div class="col-md-3">Name:</div>
15          <div class="col-md-6">{{product.productName}}</div>
16        </div>
17        <div class="row">
18          <div class="col-md-3">Code:</div>
19          <div class="col-md-6">{{product.productCode}}</div>
20        </div>
21        <div class="row">
22          <div class="col-md-3">Description:</div>
23          <div class="col-md-6">{{product.description}}</div>
24        </div>
25        <div class="row">
26          <div class="col-md-3">Availability:</div>
27          <div class="col-md-6">{{product.releaseDate}}</div>
28        </div>
29        <div class="row">
30          <div class="col-md-3">Price:</div>
31          <div class="col-md-6">{{product.price|currency:"USD": "symbol"}}</div>
32        </div>
33        <div class="row">
34          <div class="col-md-3">5 Star Rating:</div>
35          <div class="col-md-6">
36            <pm-star [rating]="product.starRating">
37            </pm-star>
38          </div>
39        </div>
40        <div class="row">
41          <div class="col-md-3">Tags:</div>
42          <div class="col-md-6">{{product.tags}}</div>
43        </div>
44      </div>
45
46      <div class="col-md-4"
47        *ngIf="product.imageUrl">
48        <img class="center-block img-responsive"
49          [style.height.px] = "150"
50          [style.margin.px] = "2"
51          [src] = "product.imageUrl"
52          [title] = "product.productName">
53      </div>
54    </div>
55
56    <div class="row mt-4">
57      <div class="col-md-4">
58        <button class="btn btn-outline-secondary mr-3"
59          style="width:80px"
60          (click)="onBack()">
```

```
61      <i class="fa fa-chevron-left"></i> Back
62    </button>
63    <button class="btn btn-outline-primary"
64        style="width:80px"
65        [routerLink]=["/products", product.id, 'edit']">
66      Edit
67    </button>
68  </div>
69</div>
70</div>
71</div>
72<div class="alert alert-danger"
73  *ngIf="errorMessage">{{errorMessage}}</div>
74</div>
75</div>
```

```
1 import { InMemoryDbService } from 'angular-in-memory-web-api';
2
3 import { Product } from './product';
4
5 export class ProductData implements InMemoryDbService {
6
7   createDb() {
8     const products: Product[] = [
9       {
10         'id': 1,
11         'productName': 'Leaf Rake',
12         'productCode': 'GDN-0011',
13         'releaseDate': 'March 19, 2018',
14         'description': 'Leaf rake with 48-inch wooden handle',
15         'price': 19.95,
16         'starRating': 3.2,
17         'imageUrl':
18           'https://openclipart.org/image/300px/svg_to_png/26215/Anonymous_Leaf_Rake.png',
19         'tags': ['rake', 'leaf', 'yard', 'home']
20       },
21       {
22         'id': 2,
23         'productName': 'Garden Cart',
24         'productCode': 'GDN-0023',
25         'releaseDate': 'March 18, 2018',
26         'description': '15 gallon capacity rolling garden cart',
27         'price': 32.99,
28         'starRating': 4.2,
29         'imageUrl':
30           'https://openclipart.org/image/300px/svg_to_png/58471/garden_cart.png'
31       },
32       {
33         'id': 5,
34         'productName': 'Hammer',
35         'productCode': 'TBX-0048',
36         'releaseDate': 'May 21, 2018',
37         'description': 'Curved claw steel hammer',
38         'price': 8.9,
39         'starRating': 4.8,
40         'imageUrl':
41           'https://openclipart.org/image/300px/svg_to_png/73/rejon_Hammer.png',
42         'tags': ['tools', 'hammer', 'construction']
43       },
44       {
45         'id': 8,
46         'productName': 'Saw',
47         'productCode': 'TBX-0022',
48         'releaseDate': 'May 15, 2018',
49         'description': '15-inch steel blade hand saw',
50         'price': 11.55,
51         'starRating': 3.7,
52         'imageUrl':
53           'https://openclipart.org/image/300px/svg_to_png/27070/egore911_saw.png'
54       },
55       {
56         'id': 10,
57         'productName': 'Video Game Controller',
58         'productCode': 'GMG-0042',
59         'releaseDate': 'October 15, 2018',
60         'description': 'Standard two-button video game controller',
61       }
62     ];
63   }
64 }
```

```
57     'price': 35.95,
58     'starRating': 4.6,
59     'imageUrl': 'https://openclipart.org/image/300px/svg_to_png/120337/xbox-
controller_01.png'
60   }
61 ];
62 return { products };
63 }
64 }
65 }
```

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   templateUrl: './welcome.component.html'
5 })
6 export class WelcomeComponent {
7   public pageTitle = 'Welcome';
8 }
9
```

```
1 <div class="card">
2   <div class="card-header">
3     {{pageTitle}}
4   </div>
5   <div class="card-body">
6     <div class="container-fluid">
7       <div class="text-center">
8         
11      </div>
12
13      <div class="text-center">Developed by:</div>
14      <div class="text-center">
15        <h3>Deborah Kurata</h3>
16      </div>
17
18      <div class="text-center">@deborahkurata</div>
19      <div class="text-center">
20        <a href="http://www.bit.ly/DeborahKsBlog">www.bit.ly/DeborahKsBlog</a>
21      </div>
22    </div>
23  </div>
24 </div>
```