

Services and Dependency Injection

523419

(Advanced Web Application Development)

Dr. Nuntawut Kaoungku
Assistant Professor of Computer Engineering

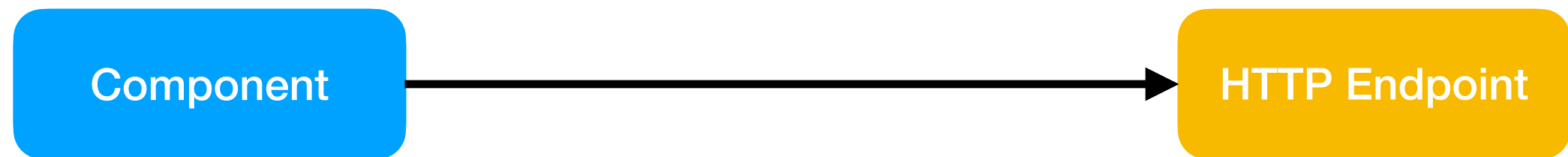
What is a Service?

- Services are singleton classes which can be injected into Angular components
- Makes it easy to create re-usable method and functionality
- Data can be easily shared between various applications Angular components using Services
- Good practice to keep all HTTP calls for a feature inside a single service file
- No need to inject in `app.module.ts` file

What is a Service? (contd.)

- Generate Angular services using Angular CLI command
 - `ng generate service <service_name>` or
 - `ng g s <service_name>`
- Inject in components where only required
- Components can have any number of services injected in the components

Code Reusability



Code Reusability (contd.)

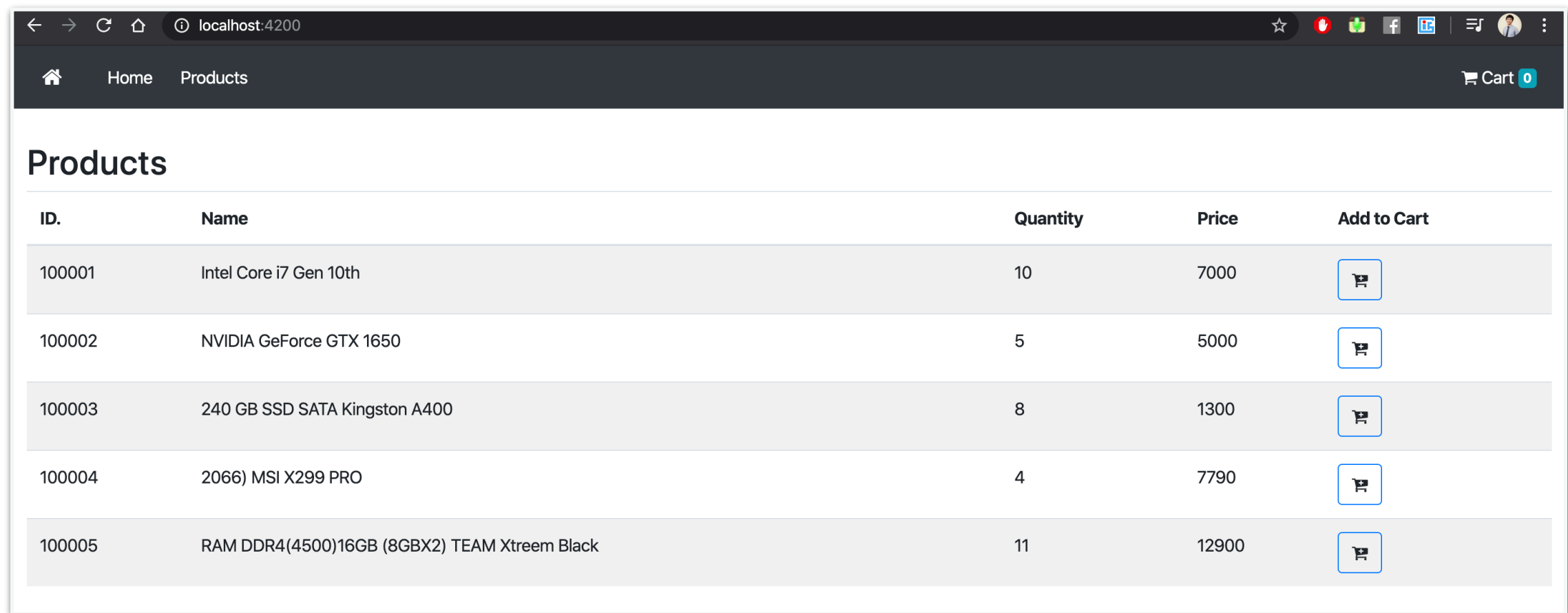
```
<> products.component.html ×
lab6-angular-app > src > app > components > products > <> products.component.html > ...
1  <h2>Products</h2>
2  <div class="table-responsive-md">
3  <table class="table table-striped table-inverse">
4    <thead class="thead-inverse">
5      <tr>
6        <th>ID.</th>
7        <th>Name</th>
8        <th>Quantity</th>
9        <th>Price</th>
10       <th>Add to Cart</th>
11     </tr>
12   </thead>
13   <tbody>
14     <tr *ngFor="let item of getAllProduct(); index as i">
15       <td scope="row">{{item.p_id}}</td>
16       <td>{{item.p_name}}</td>
17       <td>{{item.p_quantity}}</td>
18       <td>{{item.p_price}}</td>
19       <td>
20         <a class="btn btn-outline-primary" role="button">
21           <i class="fa fa-cart-plus" aria-hidden="true"></i></a>
22       </td>
23     </tr>
24   </tbody>
25 </table>
26 </div>
27
```






Template

```
TS products.component.ts ×
lab6-angular-app > src > app > components > products > TS products.component.ts > ProductsComponent > product_list
1  import { Component, OnInit } from '@angular/core';
2  import { productsType } from '../products.model'
3
4  @Component({
5    selector: 'app-products',
6    templateUrl: './products.component.html',
7    styleUrls: ['./products.component.css'],
8  })
9
10 export class ProductsComponent implements OnInit {
11
12   product_list: productsType = [
13     {p_id:100001, p_name:'Intel Core i7 Gen 10th', p_quantity:10, p_price:7000 },
14     {p_id:100002, p_name: 'NVIDIA GeForce GTX 1650', p_quantity: 5, p_price:5000 },
15     {p_id:100003, p_name: '240 GB SSD SATA Kingston A400', p_quantity: 8, p_price:1300 },
16     {p_id:100004, p_name: '2066) MSI X299 PRO', p_quantity: 4, p_price:7790 },
17     {p_id:100005, p_name: 'RAM DDR4(4500)16GB (8GBX2) TEAM Xtrem Black', p_quantity: 11, p_price:12900 }
18   ]
19
20   constructor() { }
21
22   ngOnInit(): void {}
23
24   getAllProducts(){
25     return this.product_list;
26   }
27
28 }
```

Controller

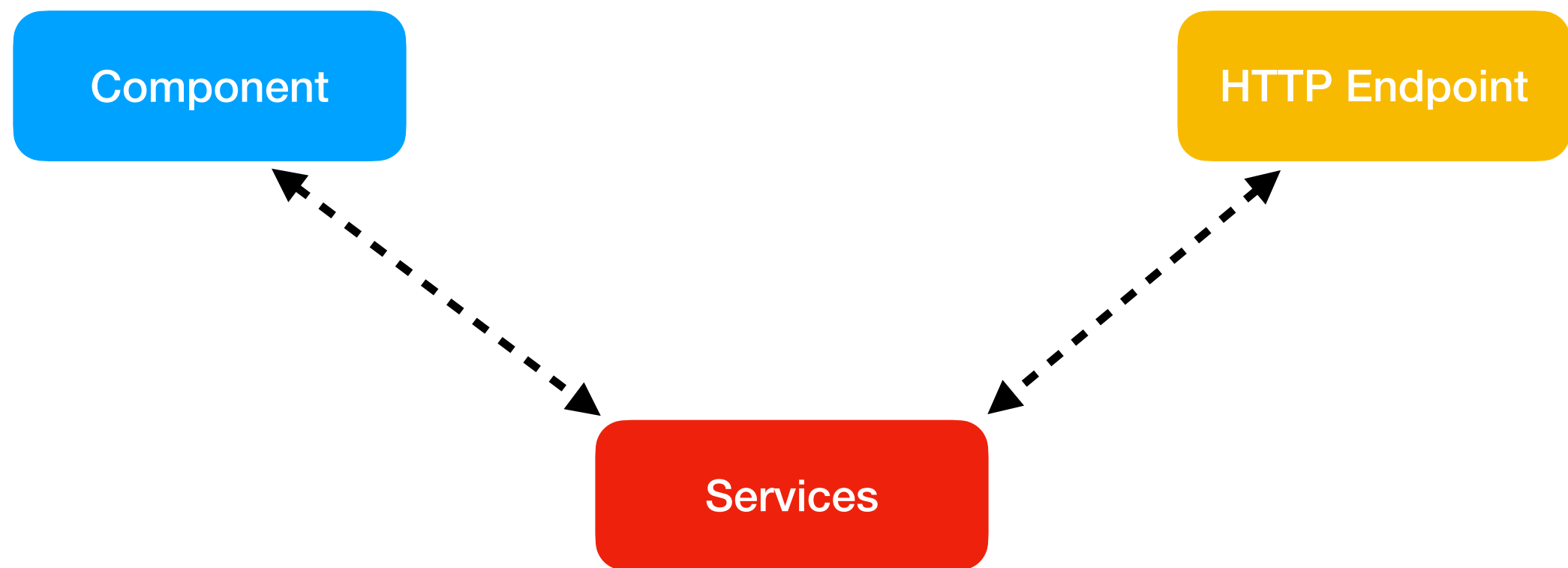
Code Reusability (contd.)



ID.	Name	Quantity	Price	Add to Cart
100001	Intel Core i7 Gen 10th	10	7000	
100002	NVIDIA GeForce GTX 1650	5	5000	
100003	240 GB SSD SATA Kingston A400	8	1300	
100004	2066) MSI X299 PRO	4	7790	
100005	RAM DDR4(4500)16GB (8GBX2) TEAM Xtrem Black	11	12900	

Display

Creating a Service



Creating a Service

TS products.service.ts ×



lab6-angular-app > src > app > services > TS products.service.ts > ...

```
1  import { Injectable } from '@angular/core';
2  import {productsType} from '../products.model'
3
4  @Injectable()
5  export class ProductsService {
6
7      product_list: productsType = [
8          {p_id:100001, p_name:'Intel Core i7 Gen 10th', p_quantity:10, p_price:7000 },
9          {p_id:100002, p_name: 'NVIDIA GeForce GTX 1650', p_quantity: 5, p_price:5000 },
10         {p_id:100003, p_name: '240 GB SSD SATA Kingston A400', p_quantity: 8, p_price:1300 },
11         {p_id:100004, p_name: '2066) MSI X299 PRO', p_quantity: 4, p_price:7790 },
12         {p_id:100005, p_name: 'RAM DDR4(4500)16GB (8GBX2) TEAM Xtream Black', p_quantity: 11, p_price:12900 }
13     ]
14
15     constructor() { }
16
17     getAllProduct(){
18         return this.product_list;
19     }
20
21 }
22
```


Using a Service

<> products.component.html X

lab6-angular-app > src > app > components > products > <> products.component.html > ...

```
1 <h2>Products</h2>
2 <div class="table-responsive-md">
3 <table class="table table-striped table-inverse">
4   <thead class="thead-inverse">
5     <tr>
6       <th>ID.</th>
7       <th>Name</th>
8       <th>Quantity</th>
9       <th>Price</th>
10      <th>Add to Cart</th>
11    </tr>
12  </thead>
13  <tbody>
14    <tr *ngFor="let item of getAllProduct(); index as i">
15      <td scope="row">{{item.p_id}}</td>
16      <td>{{item.p_name}}</td>
17      <td>{{item.p_quantity}}</td>
18      <td>{{item.p_price}}</td>
19      <td>
20        <a class="btn btn-outline-primary" role="button">
21          <i class="fa fa-cart-plus" aria-hidden="true"></i></a>
22      </td>
23    </tr>
24  </tbody>
25 </table>
26 </div>
27
```

Template

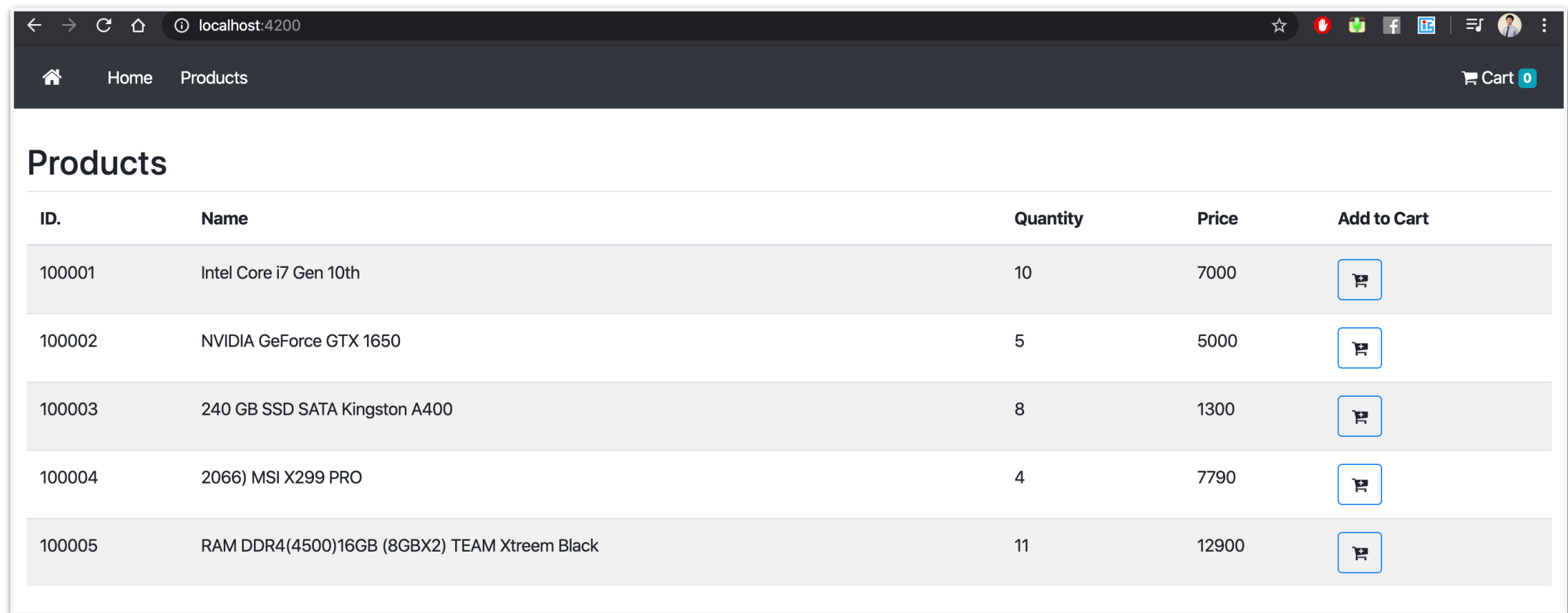
TS products.component.ts •






lab6-angular-app > src > app > components > products > TS products.component.ts > ...

```
1 import { Component, OnInit } from '@angular/core';
2 import { ProductService } from '../../services/products.service'
3
4 @Component({
5   selector: 'app-products',
6   templateUrl: './products.component.html',
7   styleUrls: ['./products.component.css'],
8 })
9
10 export class ProductsComponent implements OnInit {
11
12   constructor(private productService:ProductService) {
13
14   }
15
16   ngOnInit(): void {
17
18   }
19
20   getAllProduct(){
21     return this.productService.getAllProduct();
22   }
23
24 }
```

Controller

Displaying the component



Products				
ID.	Name	Quantity	Price	Add to Cart
100001	Intel Core i7 Gen 10th	10	7000	
100002	NVIDIA GeForce GTX 1650	5	5000	
100003	240 GB SSD SATA Kingston A400	8	1300	
100004	2066) MSI X299 PRO	4	7790	
100005	RAM DDR4(4500)16GB (8GBX2) TEAM Xtream Black	11	12900	

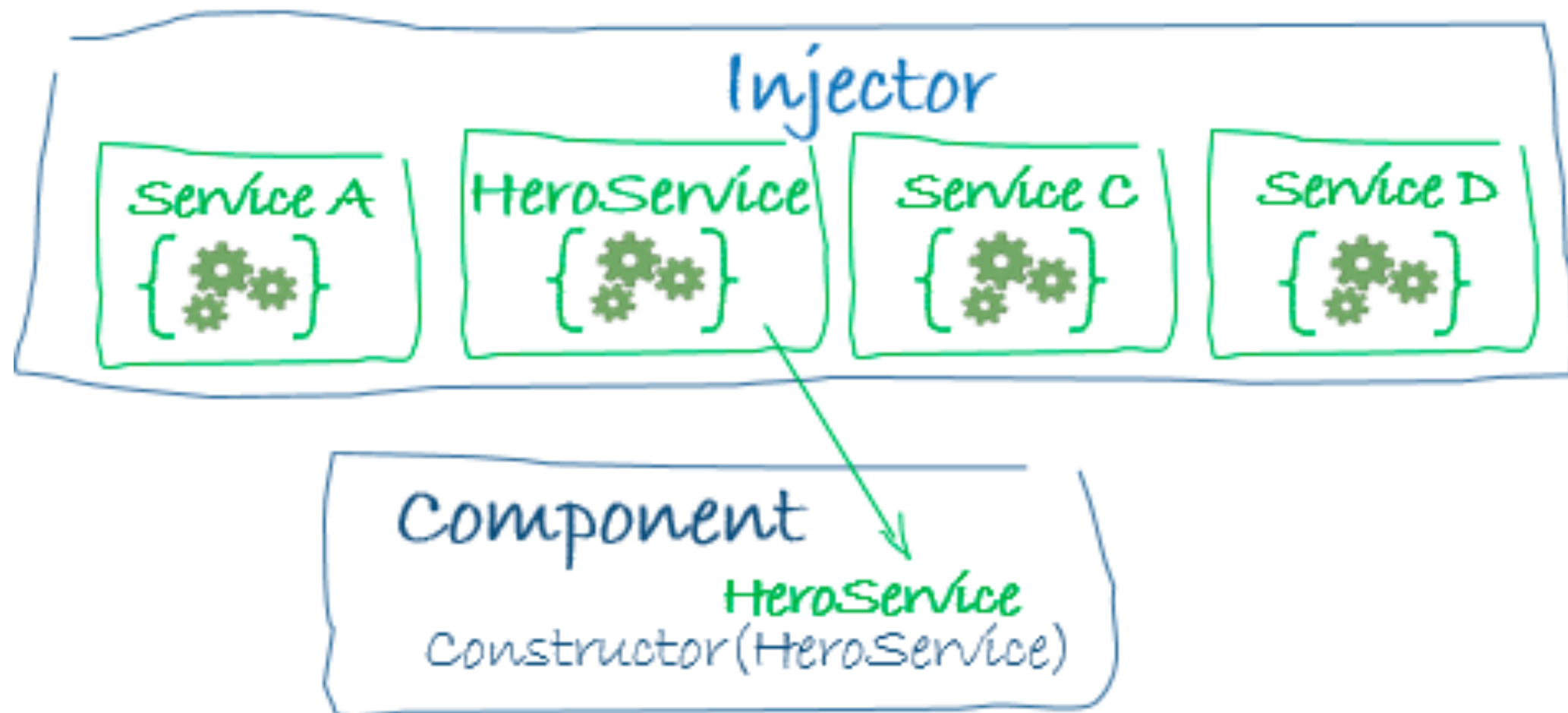
Dependency injection (DI)

- DI is wired into the Angular framework and used everywhere to provide new components with the services or other things they need.
- Components consume services; that is, you can inject a service into a component, giving the component access to that service class.
- To define a class as a service in Angular, use the `@Injectable()` decorator to provide the metadata that allows Angular to inject it into a component as a dependency.

Dependency injection (contd.)

- The injector is the main mechanism. Angular creates an application-wide injector for you during the bootstrap process, and additional injectors as needed. You don't have to create injectors.
- An injector creates dependencies, and maintains a container of dependency instances that it reuses if possible.
- A provider is an object that tells an injector how to obtain or create a dependency.

Dependency injection (contd.)



Injector & Provider

- You can configure injectors with providers at different levels of your app, by setting a metadata value in one of three places:
 - In the `@Injectable()` decorator for the service itself.
 - In the `@NgModule()` decorator for an `NgModule`.
 - In the `@Component()` decorator for a component.

Injecting services

- You can tell Angular to inject a dependency in a component's constructor by specifying a constructor parameter with the dependency type.

```
export class ProductsComponent implements OnInit {  
  constructor(private productService:ProductsService) {  
  
  }  
}
```

@Injectable Provider

- When you provide the service at the root level, Angular creates a single, shared instance of ProductService and injects it into any class that asks for it.
- Registering the provider in the @Injectable() metadata also allows Angular to optimize an app by removing the service from the compiled app if it isn't used.

TS products.service.ts ✕

lab6-angular-app > src > app > services > TS products.service.ts > ProductsService >

```
1  import { Injectable } from '@angular/core';
2  import {productsType} from '../products.model'
3
4  @Injectable({
5    providedIn: "root"
6  })
7  export class ProductService {
8
```


@NgModule Provider

- When you register a provider with a specific NgModule, the same instance of a service is available to all components in that NgModule.
- To register at this level, use the providers property of the @NgModule() decorator,

```
TS app.module.ts •
lab6-angular-app > src > app > TS app.module.ts > ...
1  import { BrowserModule } from '@angular/platform-browser';
2  import { NgModule } from '@angular/core';
3
4  import { AppRoutingModule } from './app-routing.module';
5  import { AppComponent } from './app.component';
6  import { MenuComponent } from './components/menu/menu.component';
7  import { ProductsComponent } from './components/products/products.component';
8
9  import { ProductsService } from './services/products.service'
10
11  @NgModule({
12    declarations: [
13      AppComponent,
14      MenuComponent,
15      ProductsComponent
16    ],
17    imports: [
18      BrowserModule,
19      AppRoutingModule
20    ],
21    providers: [ProductsService],
22    bootstrap: [AppComponent]
23  })
24  export class AppModule { }
25
```

@Component Provider

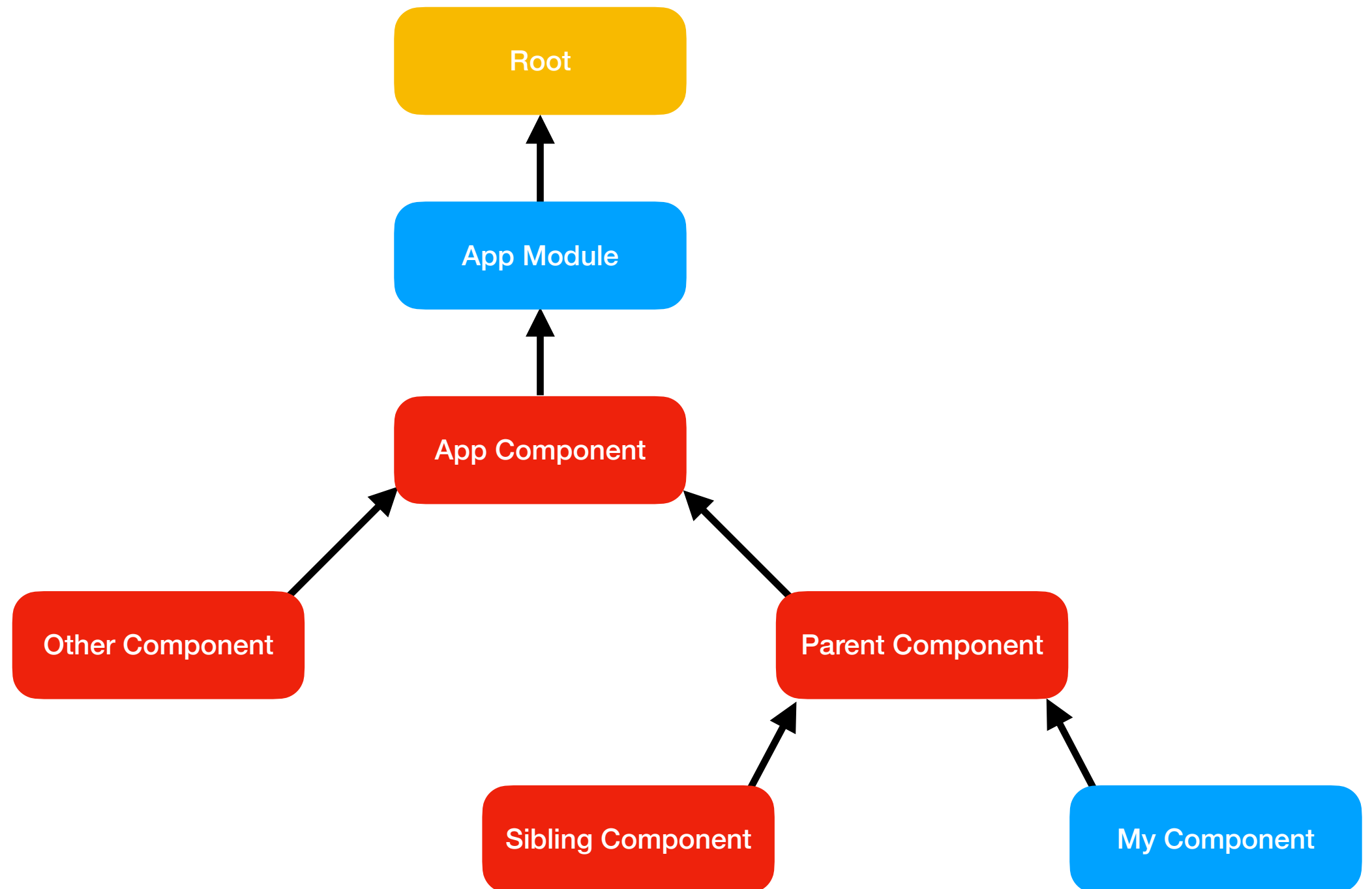
- When you register a provider at the component level, you get a new instance of the service with each new instance of that component.
- At the component level, register a service provider in the providers property of the @Component() metadata.

TS products.component.ts ●

lab6-angular-app > src > app > components > products > TS products.component.ts >  ProductsComponent

```
1  import { Component, OnInit } from '@angular/core';
2  import { ProductsService } from '../services/products.service'
3
4  @Component({
5    selector: 'app-products',
6    templateUrl: './products.component.html',
7    styleUrls: ['./products.component.css'],
8    providers: [ProductsService]
9  })
10
```

Hierarchical injector



Services that need other services

TS cart.service.ts •

lab6-angular-app > src > app > services > TS cart.service.ts > ...

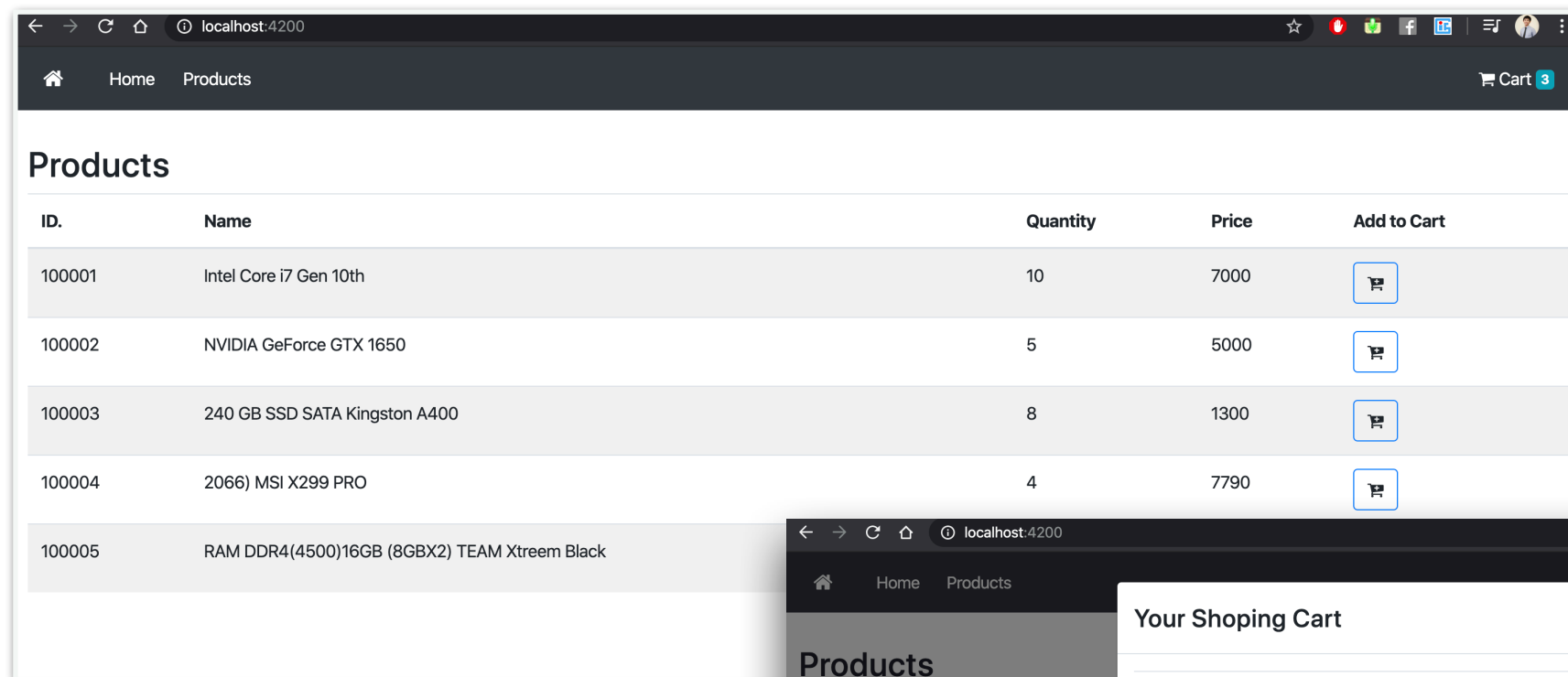
```
1  import { Injectable } from '@angular/core';
2  import { productsType } from '../products.model'
3  import { ProductsService } from '../products.service'
4
5  @Injectable()
6  export class CartService {
7
8      counter: number = 0 ;
9      sumPrice: number = 0 ;
10     cart: productsType = []
11
12     constructor(private productService: ProductsService) { }
13
14     add(p_id: number){
15         this.cart.push(this.productService.getSomeProduct(p_id));
16         this.counter = this.cart.length;
17         this.sumPrice += this.productService.getSomeProduct(p_id).p_price
18     }
19
20     getCounter(){
21         return this.counter;
22     }
23
24     getsumPrice(){
25         return this.sumPrice;
26     }
27
28     getCart(){
29         return this.cart;
30     }
31 }
32
```





TS products.service.ts •

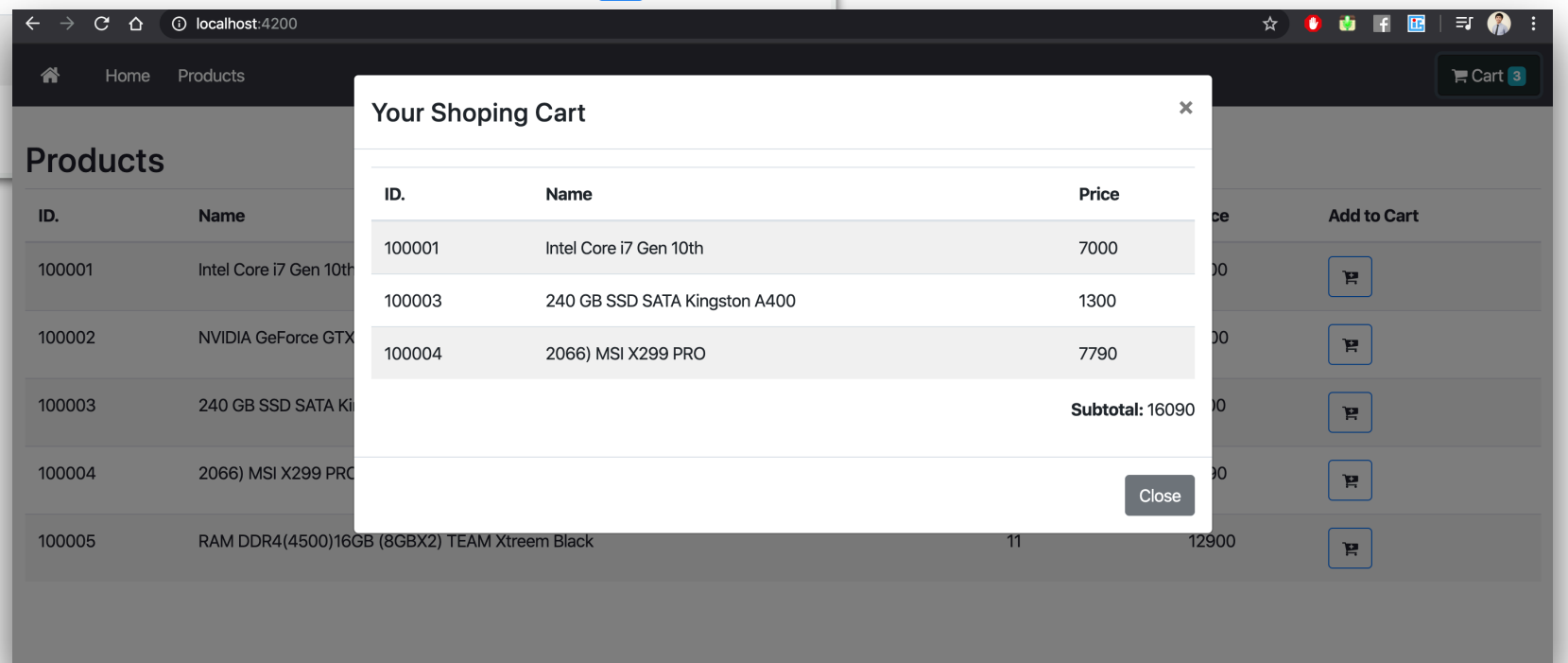
lab6-angular-app > src > app > services > TS products.service.ts > ...

```
1  import { Injectable } from '@angular/core';
2  import { productsType } from '../products.model'
3
4  @Injectable()
5  export class ProductsService {
6
7      > product_list: productsType = [ ...
13     ]
14
15     constructor() { }
16
17     getAllProduct(){
18         return this.product_list;
19     }
20
21     getSomeProduct(p_id: number){
22         return this.product_list[p_id]
23     }
24 }
25
```

Displaying the component



ID.	Name	Quantity	Price	Add to Cart
100001	Intel Core i7 Gen 10th	10	7000	
100002	NVIDIA GeForce GTX 1650	5	5000	
100003	240 GB SSD SATA Kingston A400	8	1300	
100004	2066) MSI X299 PRO	4	7790	
100005	RAM DDR4(4500)16GB (8GBX2) TEAM Xtrem Black			



ID.	Name	Price
100001	Intel Core i7 Gen 10th	7000
100003	240 GB SSD SATA Kingston A400	1300
100004	2066) MSI X299 PRO	7790
		Subtotal: 16090

Close