



Angular Services

+

Angular Service: Generate new Service

Angular service berisikan method/function untuk manage data yang berasal dari server.

Untuk men-generate file service via cli, gunakan command

```
ng generate service nama_service
```

Dalam sesi ini, kita akan membuat 1 service untuk product list.

```
ng generate service product
```

Angular Service: Generate new Service

```
product.service.ts U ●
src > app > product.service.ts > ...
1  import { Injectable } from '@angular/core';
2
3  @Injectable({
4    providedIn: 'root'
5  })
6  export class ProductService {
7
8    constructor() { }
9  }
```

File product.service.ts di samping adalah contoh file service yang di-generate.

@Injectable merupakan decorator yang menandai bahwa class service dapat dipakai/di-inject sebagai dependency dimanapun dibutuhkan di dalam aplikasi angular.

Angular Service: Mock Products

Buatlah file product.ts dan mock-products.ts yang berisi sebagai berikut:

Product memiliki 3 properti, yaitu id, name, dan description

TS product.ts U X

src > app > TS product.ts > ...

```
1  export interface Product {  
2      id: string;  
3      name: string;  
4      description: string;  
5  }
```

TS mock-products.ts U X

src > app > TS mock-products.ts > ...

```
1  import { Product } from './product';  
2  
3  export const PRODUCTS: Product[] = [  
4      {  
5          id: "ABC8441189035",  
6          name: "Tshirt",  
7          description: "Lorem ipsum dolor sit amet, consectetur adipiscing elit."  
8      },  
9      {  
10         id: "DEF6510463347",  
11         name: "Shoes",  
12         description: "Proin ac metus in diam porttitor viverra eu sit amet ligula."  
13     },  
14     {  
15         id: "GHI0831819467",  
16         name: "Handbags",  
17         description: "Duis sodales dui vitae urna varius, at ullamcorper purus tempor."  
18     }  
19 ]
```



Angular Service: Pembuatan method di service

```
product.service.ts U ×
src > app > product.service.ts > ...
1  import { Injectable } from '@angular/core';
2
3  import { Product } from './product'
4  import { PRODUCTS } from './mock-products'
5  import { Observable, of } from 'rxjs';
6
7  @Injectable({
8    providedIn: 'root'
9  })
10 export class ProductService {
11
12   constructor() { }
13
14   getProducts(): Observable<Product[]> {
15     const products = of(PRODUCTS);
16
17     console.log("fetched list of products....");
18     return products;
19   }
20 }
21
```

- Import file product, mock-product dan Observable serta of dari rxjs.
- Buat 1 method di file product.service untuk mendapatkan list product dari file mock-products. Contoh method di samping bernama **getProducts**
- **Observable** menyediakan layanan sharing data antar publisher dan subscriber agar dapat berkomunikasi antar komponen. getProducts disini akan bertugas menjadi **publisher**.
- **of()** merupakan method yang disediakan rxjs, agar PRODUCTS menjadi fake response dari Observable.

Angular Service: Import service di komponen

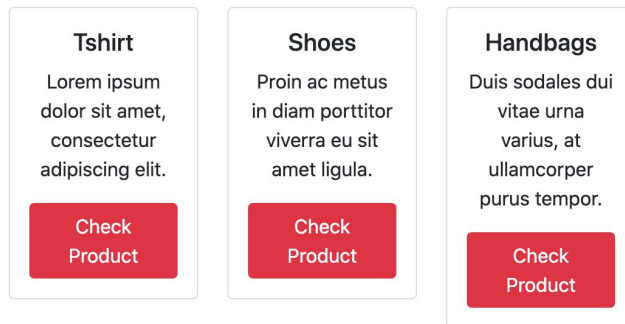
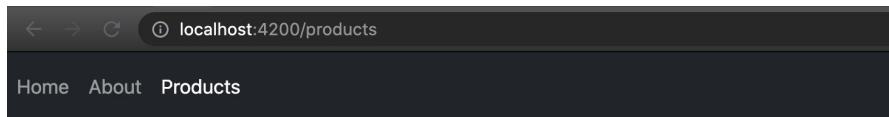
```
products.component.ts U x
src > app > products > products.component.ts > ... 1
1 import { Component, OnInit } from '@angular/core';
2 import { Product } from '../product'
3 import { ProductService } from '../product.service';
4
5 > @Component({ ...
9 })
10 export class ProductsComponent implements OnInit {
11   products: Product[] = []; 2
12
13   constructor(private productService: ProductService) { } 3
14
15   ngOnInit() {
16     this.getProducts(); 5
17   }
18
19   getProducts() {
20     this.productService 4
21     .getProducts()
22     .subscribe(products => this.products = products);
23   }
24 }
```

1. Import **Product** dan **ProductService**
2. Buat properti **products**, yang bertipekan Product Array. Value awal merupakan array kosong.
3. kita akan meng-inject service yang dibuat sehingga bisa digunakan dalam komponen, dengan memasukkan parameter **productService** yang bertipekan **ProductService** kedalam constructor.
4. Buatlah method **getProducts**, yang akan men-**subscribe** method **getProducts** dari **productService**. Serta memasukkan response dari service ke properti **products** yang sebelumnya dibuat.
5. Di method hooks **ngOnInit**, panggil method **this.getProducts** yang sebelumnya dibuat, agar service selalu dipanggil saat halaman products di reload.



Angular Service: Preview

```
products.component.html U X
src > app > products > products.component.html > ...
1 <div class="row">
2   <div class="col-sm-4" *ngFor="let product of products">
3     <div class="card mb-3">
4       <div class="card-body">
5         <h5 class="card-title">{{product.name}}</h5>
6         <p class="card-text">{{product.description}}</p>
7         <a [routerLink]="['/product-detail/', product.id]" class="btn btn-danger">
8           Check Product
9         </a>
10      </div>
11    </div>
12  </div>
13 </div>
```



Jika sudah, maka akses ke path `/products`. Maka akan muncul list products yang berasal dari service