

Angular Services

Creating Custom Angular Services



Jim Cooper

Software Engineer

@jimthecoop | jcoop.io



Version Check



This course was created by using:
- Angular 17



Version Check



This course is 100% applicable to:

- Angular 14+

Mostly applicable to:

- 8+



Recommended Prerequisites

Angular Fundamentals

HTML

JavaScript



Q: What are Angular Services?

A: Simply a TypeScript Class



Services Are Just Classes

```
class ProductService {  
  
}
```



Services Are Just Classes

```
class ProductService {  
    getProduct(id:string): Observable<Product> {  
        return ...;  
    }  
}
```



Q: Why are services necessary?



Why are Services Necessary?

Encapsulation

Code reuse

Separation of concerns

Dependency injection



Services Are Just Classes

```
class ProductService {  
  
    getProduct(id:string): Observable<Product> {  
        return httpClient.get(...);  
    }  
  
}
```



Services Are Just Classes

```
class ProductService {  
  
    getProduct(id:string): Observable<Product> {  
        const httpClient:HttpClient = new httpClient(...);  
        return httpClient.get(...);  
    }  
  
}
```



Services Are Just Classes

```
class ProductService {  
    constructor(private httpClient: HttpClient) { }  
  
    getProduct(id:string): Observable<Product> {  
        return httpClient.get(...);  
    }  
}
```



Why are Services Necessary?

Encapsulation

Code reuse

Separation of concerns

Dependency injection










Single Responsibility Principle



Single Responsibility Principle



<div>  <div> <div>JOE'S</div> <div>ROBOT</div> <div>SHOP</div> </div> </div> <div> Catalog Search Cart </div>		<div> Sign In Register </div>	
<div>  <div> <div>Large Cyclops</div> <div>A robot head with an unusually large eye and teloscopic neck -- excellent for exploring high spaces.</div> <div>Part Type: Heads</div> </div> </div>		<div> <div> <div>\$1,220.50</div> <div>\$976.40</div> </div> <div>Buy</div> </div>	
<div>  <div> <div>Spring Base</div> <div>A spring base - great for reaching high places.</div> <div>Part Type: Bases</div> </div> </div>		<div> <div> <div>\$1,190.50</div> </div> <div>Buy</div> </div>	
<div>  <div> <div>Articulated Arm</div> <div>An articulated arm with a claw -- great for reaching around corners or working in tight spaces.</div> <div>Part Type: Arms</div> </div> </div>		<div> <div> <div>\$275.00</div> </div> <div>Buy</div> </div>	
<div>  <div> <div>Friendly Bot</div> <div>A friendly robot head with two eyes and a smile -- great for domestic use.</div> <div>Part Type: Heads</div> </div> </div>		<div> <div> <div>\$945.00</div> <div>\$756.00</div> </div> <div>Buy</div> </div>	
<div>  <div> <div>Shredder</div> <div>A large three-eyed head with a shredder for a mouth -- great for crushing light medals or shredding documents.</div> <div>Part Type: Heads</div> </div> </div>		<div> <div> <div>\$1,275.50</div> </div> <div>Buy</div> </div>	
<div>  <div> <div>Single Wheeled Base</div> <div>A single-wheeled base with an accelerometer capable of higher speeds and navigating rougher terrain than the two-wheeled variety.</div> <div>Part Type: Bases</div> </div> </div>		<div> <div> <div>\$1,190.50</div> <div>\$1,071.45</div> </div> <div>Buy</div> </div>	

The Single Responsibility Principle

```
@Component({...})
export class CatalogComponent {
  products: Product[] = [];

  ngOnInit() {
    this.products = this.fetchProducts();
 }
```

```
  addToCart(product) {
    this.addProductToCart(product);
  }
```

```
  async fetchProducts() {
    return await fetch('/api/products').json();
  }
```

```
  addProductToCart(product) {
    ...
  }
```

```
<ul class="products">
  <li *ngFor="let product of
products">
    <div (click)="addToCart()">
      </div>
    <li>
  </ul>
</div>
```



The Single Responsibility Principle

```
@Component({...})
export class CatalogComponent {
  products: Product[] = [];

  ngOnInit() {
    this.products = productsService.getProducts();
  }
}
```

```
addToCart(product) {
  this.addProductToCart(product);
}
```

```
addProductToCart(product) {
  ...
}
```

```
}
```

```
<ul class="products">
  <li *ngFor="let product of
products">
    <div (click)="addToCart()">
      </div>
    <li>
  </ul>
</div>
```



The Single Responsibility Principle

```
@Component({...})
export class CatalogComponent {
  products: Product[] = [];

  ngOnInit() {
    this.products = productService.getProducts();
  }

  addToCart(product) {
    cartService.addToCart(product);
  }
}
```

```
<ul class="products">
  <li *ngFor="let product of
products">
    <div (click)="addToCart()">
      </div>
    <li>
  </ul>
</div>
```

