

Angular – @ViewChild, @ContentChild @HostListener, @HostBinding Services

Presented by

@ViewChild()

- The **@ViewChild()** decorator is used to access child components elements within a parent component.
- It allows to interact with these child elements programmatically, making it possible to call their methods.
- Ex:
 - Employee – Parent Component
 - Salary – Child Component
- Access method related to salary property of Salary component from **Employee** component

```
@ViewChild(SalaryComponent)  
salaryComponent!: SalaryComponent; // ! is like undefined
```

** Demo



@ContentChild

- **@ContentChild** is property decorators.
- It is used to a reference to the projected content.
- Projected content *is the content that the current component receives from a parent component as object.*
 - Ex.
 - Parent – NumberComponent
 - Child - TestComponent
 - [receives a reference to the projected content , i.e., a variable's value of Parent]
- Used to access elements, components, or directives projected into the component using **<ng-content>**.

** Demo



Host Listener

- The **@HostListener** decorator in Angular allows to listen to events on the host element (keypress, mouse click) of a directive or component.
- It is particularly useful for handling DOM events in a more granular way.
- Ex: Using **@HostListener** to Listen to Key Press event
- Using **@HostListener** to Listen to **Click** event

****Demo**



Host Binding

- **@HostBinding** is a decorator that enables to bind *properties of a component* or *directive to properties of the host element*.
- It is used to set property values on the host element based on the values of the component or directive properties.

```
import { Component, HostBinding } from '@angular/core';
```

```
@HostBinding('style.color') color = 'red';
```

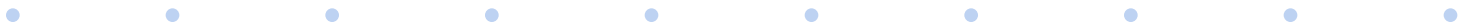
- Host element property – `style.color`
- Component property – `color`

****Demo**



Angular Services

- What is a Service?
 - Reusable code that can be accessed from multiple components
 - A service can contain a value or function or combination of both.
 - Service is a class that has the purpose of Providing a Service to a / all Component(s).
- Ex : a service provides data(data service) to every component that needs it.
- Why services are used?
 - Shares data across components
 - They are reusable.



Angular Services ...



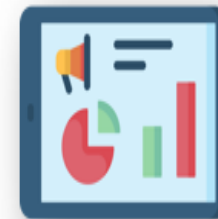
A Service is a Class



Decorated with
@Injectable



They share the same
piece of code



Hold the business logic



Interact with the
backend



Share data among
components



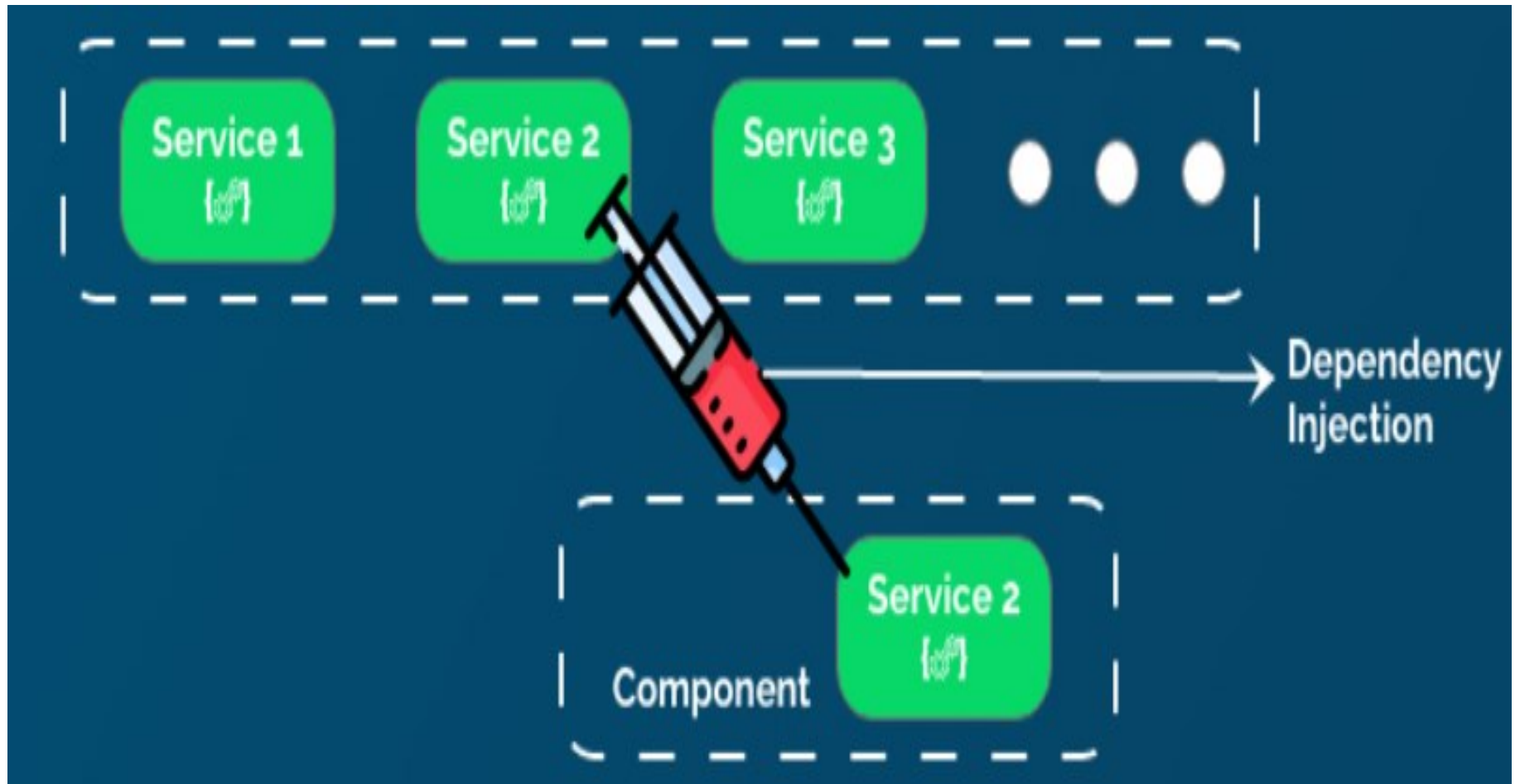
Services are singleton



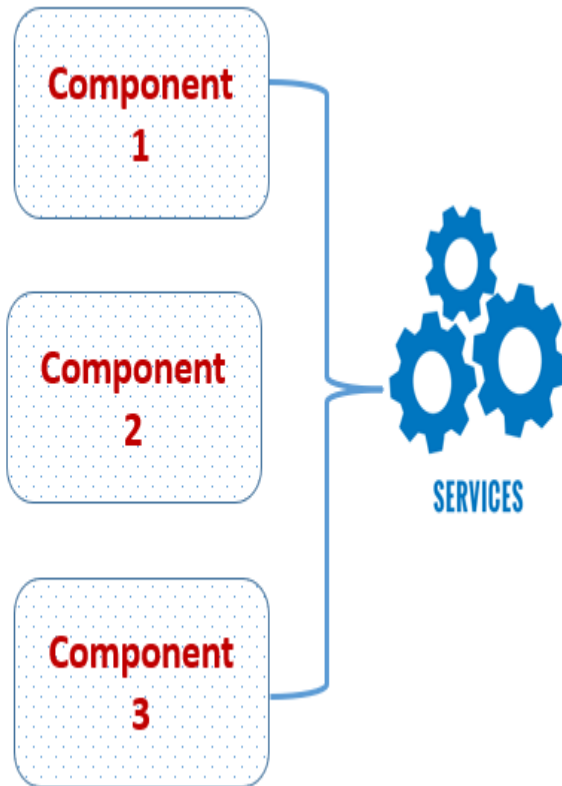
Registered on modules
or components



Dependency Injection



Services ...



ng generate service CustomerService

```
@Injectable({  
  providedIn: 'root'  
})  
export class CustomService {  
  constructor() { }  
  
  // define our custom service methods here  
}
```

```
import { CustomerService } from './customer.service'
```

** Demo

Thank You

