**✅ What is a Service in Angular? (Very Simple)**

Think of a **Service** like a **helping class** in Angular.

It is used when you want to **write some common code once** and **use it in many components**.

Service is a reusable helper class in Angular used to share data and common logic across components.

**💡 Why do we use a Service?**

* To **share data** between components
* To **write common logic** in one place (not repeat code in every component)

**🎯 Simple Meaning**

| **Without Service** | **With Service** |
| --- | --- |
| Same code written again in many components | Write code once in service and use it everywhere |
| Hard to maintain | Easy to maintain |

**⭐ Key Points (Easy to Remember)**

| **Point** | **Simple Explanation** |
| --- | --- |
| Services are singleton | Only **one object** is created and shared by all components |
| Used to share data | Example: send data from one component to another |
| Contains methods & data | We write functions and variables inside service |
| Created using a command | Angular gives a command to create a service easily |

## ✅ **What is Dependency Injection (DI)? — Very Simple**

**Dependency Injection (DI)** means **giving** the required object (like a service) to a class instead of the class **creating it by itself**.

Dependency Injection means Angular automatically provides the required service to a component instead of the component creating it.

It’s like **someone delivering the needed item to you**, so you don’t need to go get it.

| **Without DI (Bad)** | **With DI (Good)** |
| --- | --- |
| Each component creates its own objects | Objects (services) are created only once and shared |
| More code, repeated work | Less code, reusable |
| Hard to manage | Easy to manage |

### ⭐ Key Points (Easy to Remember)

| **Point** | **Simple Explanation** |
| --- | --- |
| DI is a design pattern | A way to supply (inject) required objects |
| Used in Angular | Helps components get services easily |
| Add dependency in constructor | We write service name inside constructor() to receive it |
| Injector creates objects | Angular creates and shares service object automatically |
| @Injectable needed | Marks class as a service so DI can inject it |