**Service Injection :**

* Instead of creating the service, make new, and then call it. Angular gives us dependency injection option.
* First define the target Service in the providers section. Then, add it in the constructor (cstor injection). And finally, you call it.

Une image contenant texte

Description générée automatiquement

**Hierarchical Injector :**

* When you create a new component and want to make a dependency injection, Angular creates a new instance of the service each time. So what should we do ? remove it from providers, and keep the declaration in the cstor.
* If you want the Service to be known in all your app, you should declare it in the providers at the app.module. Like this, all the components that will call your service will have the same instance of the service.

Une image contenant texte

Description générée automatiquement

**Injecting a service into a service :**

* You should add @Injectable decoration or it won’t work.
* Declare it in the constructor.
* And then call it.
* A recommendation, always add @Injectable to your service.

Une image contenant texte

Description générée automatiquement

* When you use Services, you save a lot of work, you avoid using @Input, @Output, emit events …

**Communicating between Components using Services :**

* Declare the event in the service

Une image contenant texte

Description générée automatiquement

* Emit event from another component

Une image contenant texte

Description générée automatiquement

* Subscribe to an event and listen to it from another component

Une image contenant texte

Description générée automatiquement