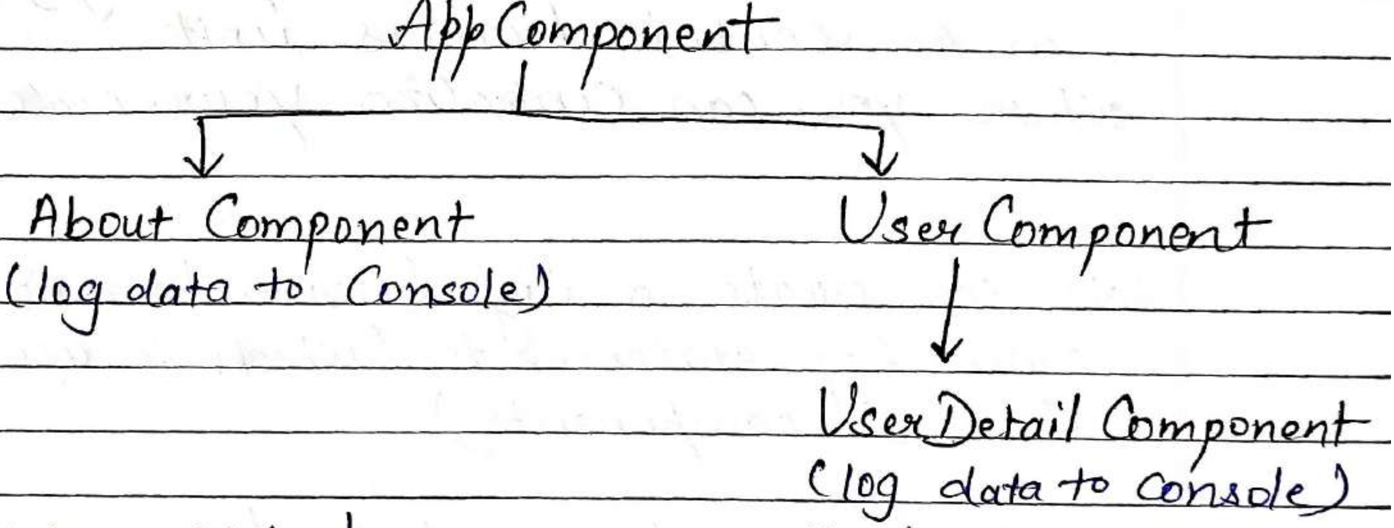
Services and Dependency Injection

Angular services are ~~sig~~ singleton object that gets instantiated only once in a lifetime of an application

Consider an application having:



We might have some methods in those components

Let's say we want to log some data in About Component and User Detail Component

Both logs are same, so basically we are duplicating code in 2 different component

Suppose in User Component, we also want to access some data, some array of users and we don't know if we will later use that in other part of application

Both User Cases

- * Duplication of Code
- * Data Storage

In both cases we, typically use Service

A Service is just another piece in your Angular App, Another class which acts as a central repository, as a central business unit where you can centralize your code

We can create a log Service to centralize our log statement (which is used in 2 components)

Service are also useful for Communication between components

How to create a Service

* Simply run command

```
ng g s service-name
```

your service will be generated

How do you Access a Service In your Component ?

Let's say you have made seervice named Logging Seervice

you can simply instantiate your Logging Service in component

account. component. ts

* ~~You can s~~ const seervice = new Logging Service();

* Import the Logging Seervice

* service. logstatus (status);
↓
your method you wanted to call from service in component.

But this is not suggested, Angular provides us more better way to Inject Service in Our Component.

Angular Dependency Injector

Angular provides us more betler way to inject service In component It is Angular dependency Injector

So, We have dependeny on logging Service from our component and dependeny injector simply injects this dependency, instance of our class automatically into our component.

We need to only Inform Angular, that we require such an instance.

How do we Inform that?

* We add constructor to the component you'll see constructor present If you have made service/ component through command

* Inside constructor, I can bind it to property

we use this access specifies (must)

Imp to specify type and that should be service

```
Constructor (private loggingService : LoggingService)
    { }
```

(make sure to import logging Service at top)

Page
Date

Now Angular know we want logging Service
But it doesn't know how to give Instance

* We need to provide a service
Add one providers property In @ Component decorator.

```
@ Component ({
    selector :
    template URL :
    Style URL :
    providers : [Logging Service] → [Here we
})                                   have provided]
```

Hierarchical Injector

Angular dependency Injector is actually a hierarchical Injector.

It means that if we provide a service in one component, then Angular framework knows how to create instance of service for this component and important, all it's child components

All of them will receive same instance of service

Hierarchical Injector

App Module → Same Instance of Service is available Application-wide

App Component → Same Instance of Service is available for all components (but not for other service)

Any Other Component ⇒ Same Instance is available for component and all its child components.

It works Top to down not down to Top

Suppose If you want to use same Instance of Service in AppComponent to child component

Then you don't use providers in child component

Everything else stays same.

Page
Date

Injecting Services into Services

The highest level of providing service. is provide your service in providers array of App.Module.TS

If we are providing service in module, we make sure we are using same instance of service everywhere, in every component throughout app Unless we are overriding our service.