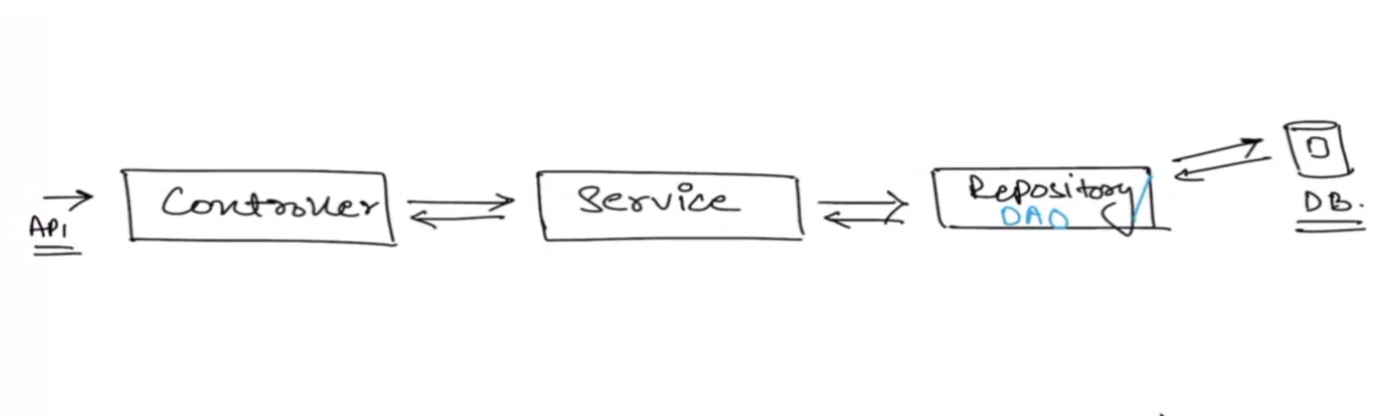
Scaler API notes: Day2



Ideally, there should be one CONTROLLER, corresponding to that there will be one SERVICE, and corresponding to that there will be one REPOSITORY.

Controller:

Controller is like a restaurant waiter; will the waiter be do the hard work (Will it have business logic)? No. It will just pass the request to Service.

When you first hit the URL, the request will first reach to the controller. and how?

Couse we’ve defined annotations.

@RESTController

@RequestMampping

* Which controller will take which kind of request will be decided by the @RequestMapping

There is a saying that “Every class in your system in your software application should be structured around the Resource (Controller) “.

Let’s say if you have product-service, order-service. And so on. Then every service will have controller.

Myth busted!

Will there be services without controller? Yes. for example:

let’s say if you have priceCalculator-service and other service which is having controller (order-service) is uses this service. But priceCalculator-service will not have the corresponding controller to it. It’s like an internal utility method.

Service :

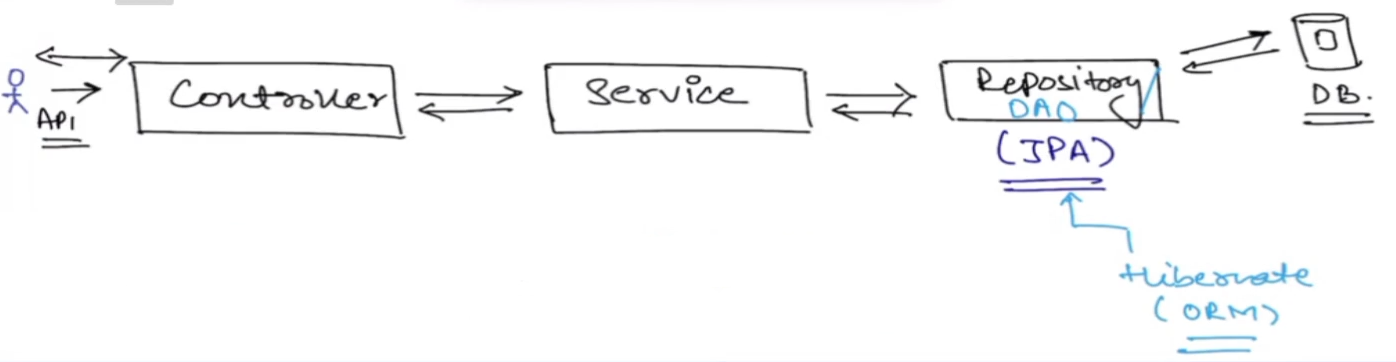
Service is out chef, he will do all the hard work (Will have the business logic)

Repository :

If Service needs to interact with database, then it will call the Repository to get the data from Database.

Repository will interact with database.

In some companies Repository is also called as DAO 🡪 Data Access Object



JPA 🡪 Java Persistence API

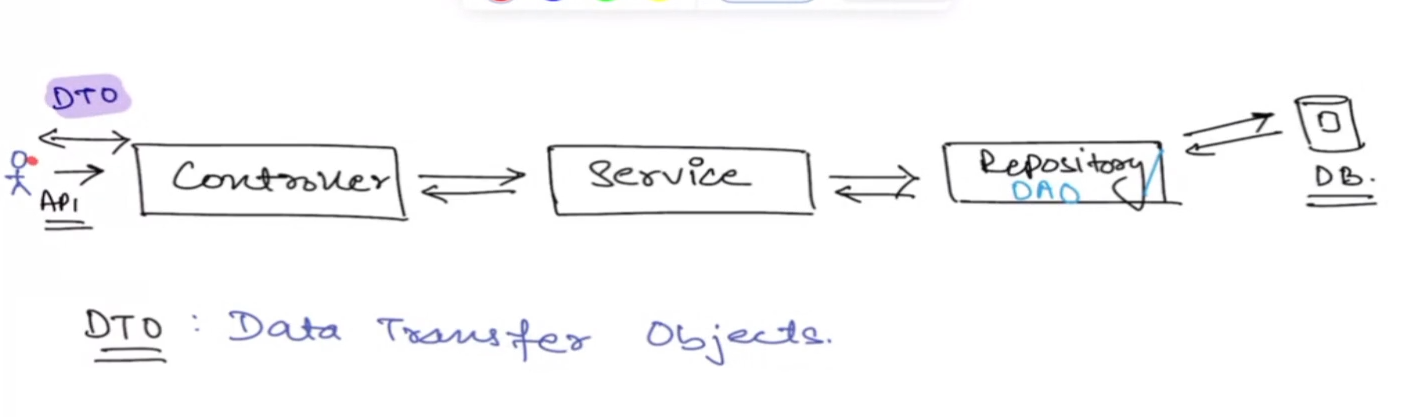
JPA is the library (Interface) provided by Spring and it is used to interact with the database. It internally implements Hibernate.

Hibernate -> It is an ORM (Object Relation Model).

Read more about it here:

<https://www.baeldung.com/the-persistence-layer-with-spring-data-jpa>Myth busted!

If you don’t have RIPOSITORY will your code still work? Yes. SERVICE can directly talk to database.



DTO 🡪 Data Transfer Object

Object that are used to transfer the data.

We can not expose all the details to the client. For example, on Amazon.in would you say that entire product information is visible to you?

No right. Amazon will only display the data which is required to you (It will not give you the inventory quantity of the particular product).

So DTO will be used at client side. DTO’s are just there to add a one extra layer of security (Privacy) from your internal system.