**Spring Boot**

**1)Annotation:**

**2)Controller vs Rest Controller**

**3)Profiling**

**4)what is setter injection vs constructor injection**

**5)what is @collifier annotations used for**

**6)what is Actuators**

**7)What is the scope of bean**

-There are mainly 5 scopes of beans in the Spring Framework

1)Singleton-(By default)

- whenever you ask for a class object or bean spring IoC container will create one single object and give it to you whenever you call throughout the application.

-it won't create a new object every time you ask for an object it gives a singleton object.

2)Prototype-

when even you ask for an object spring IOC container will create a new object and give to you

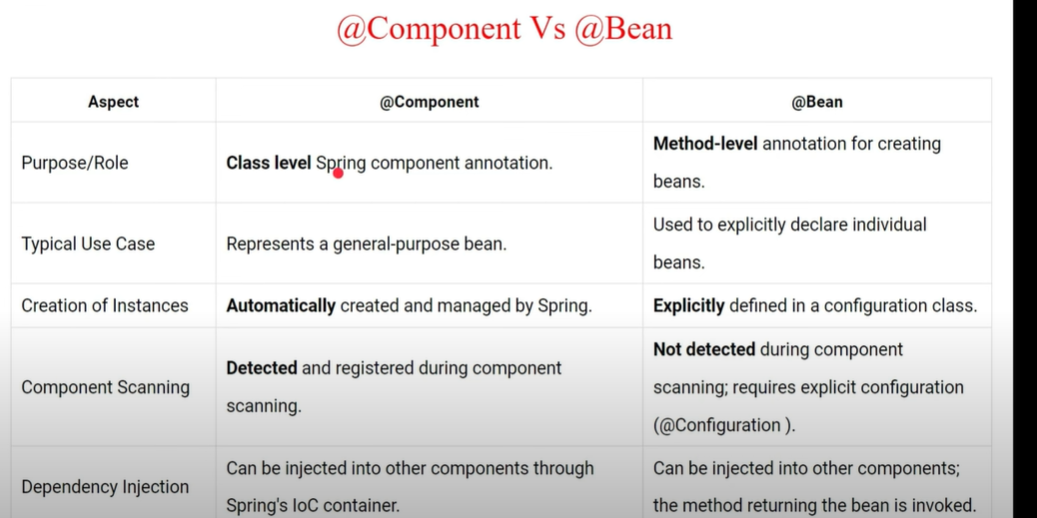
3)Request(associated with web application)

4)Session(associated with web application)

5)Global Session

**8)What is Actuators**

**9)**

****

**10)how to use two db in spring boot how we are handling that**

**11)HOW TO ENABLE SPRING ACTUATOR WHICH ARE BEDEFAULT DESIBLE**

management:

endpoints:

enabled-by-default: true

**Q. how to do Spring Boot Profiling**

We can create different applications.property file like

PROD- application-prod.properties

DEV- application-dev.properties

And in the main or default application.properties file just add one configuration like

Spring.profiles.active=dev

* We can also pass the active profile in run configuration vm argument as

-D spring.profiles.active=dev,prod

* We can have multiple profile one is from dev and one is from prod so in that case we can use **@propertySource**(“application-prod.properties”) annotataion over class so we wont get bean creation error.



* We can use @Value(“${}”) to read value from properties file to java
* We can also make sure by using @Profile(“dev”) if active profile is not dev it wont create the bean of that class

**Q. How to Solve Cycle or circular Dependency Issue in Spring boot**

-To resolve circular dependency issues in Spring Boot, you can use constructor injection, setter injection, or @Lazy annotation. Here's an example using constructor injection:

**Q. Annotation in Spring Boot**

