**Spring Boot**

**1)Annotation:**

**@SpringBootApplication:**

**@Transaction:**

**-if you have any association mapping in your class and if it fail in that case I want roll back then we use this annotation  
-suppose I have user class which has payment association class object and due to something payment got fail in such case I don’t want to save user object as payment fail then I use this**

**2)Controller vs Rest Controller**

**3)Profiling**

**@proprtySource**

**@ConfigurationProperty(prefix)**

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

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

**When I have one interface and suppose two classes providing the implemention to interface method while calling method through interface reference jvm will confuse to call which class so in that case we can use @qualifer annotation and method the class name or use @primary**

**6)what is Actuators:**

**-it provide the production ready feature like monitor, health check**

**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-

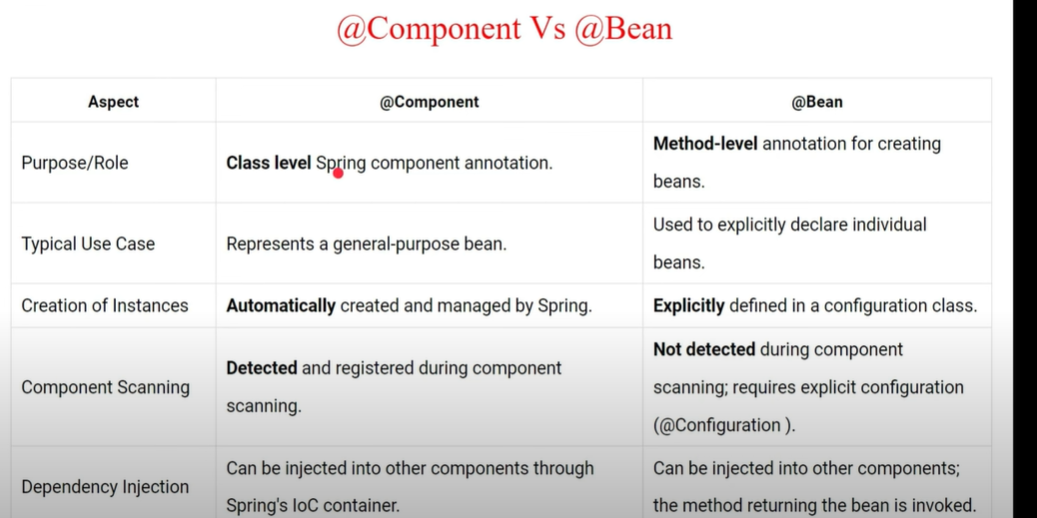
when even you ask for an object spring IOC container will create a new object and give to you

3)Request(associated with web application)

4)Session(associated with web application)

5)Global Session

**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.

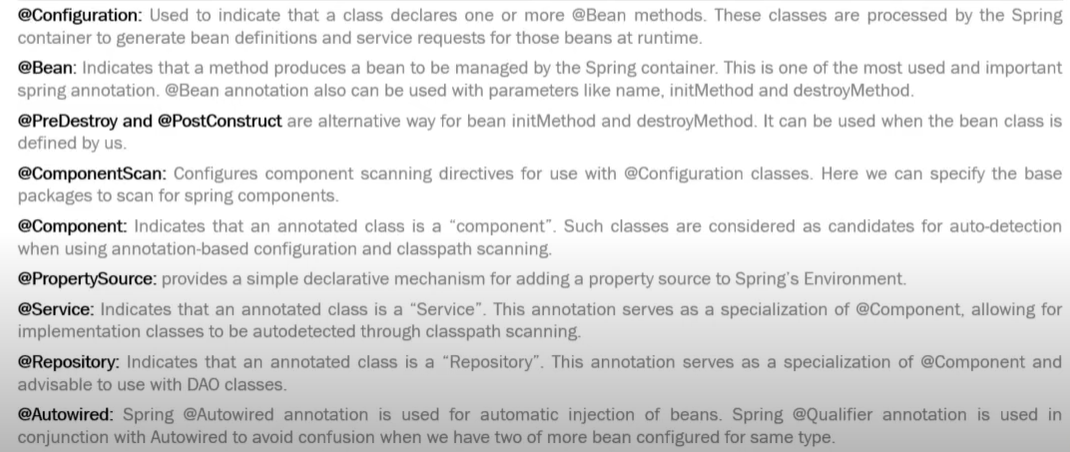


* We can use @Value(“${}”) to read value from properties file to java
* We can also make sure by using @Profile(“dev”) if active profile is not dev it wont create the bean of that class

**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**



Q. What is Demon Thread

-demon thread are low priority thread which are run at background and work on mercy of user thread. Once user thread will complete jvm won’t wait to complete demon thread task it will abnormally terminate.

-if you want to make any thread as demon thread there is method setDemon(true) before starting the thread else get illegal thread exception