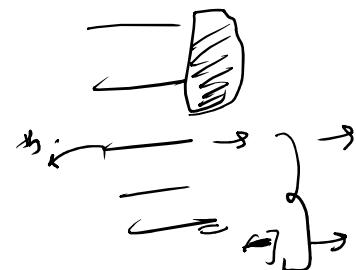


The super power of Spring

#

`Person p = new Person();`

→ Programmer was managing the life cycle of the objects



→ Spring is creating and managing the life cycle of the objects.

↳ [Inversion of Control]

① Any class which has `@Component` annotation written over it, directly/indirectly is managed by Spring.

→ Spring creates an object for all those classes which have `@Component` writers on it.

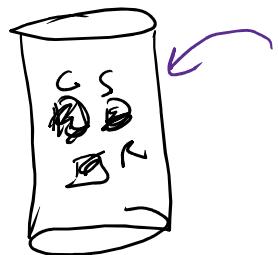
Scope : How many copies of the been created?

object here
Bean
A Java object

→ By default, scope of the bean created by Spring is 'singleton'.
A Java object created and managed by Spring.

{What all beans and config Spring is managing}

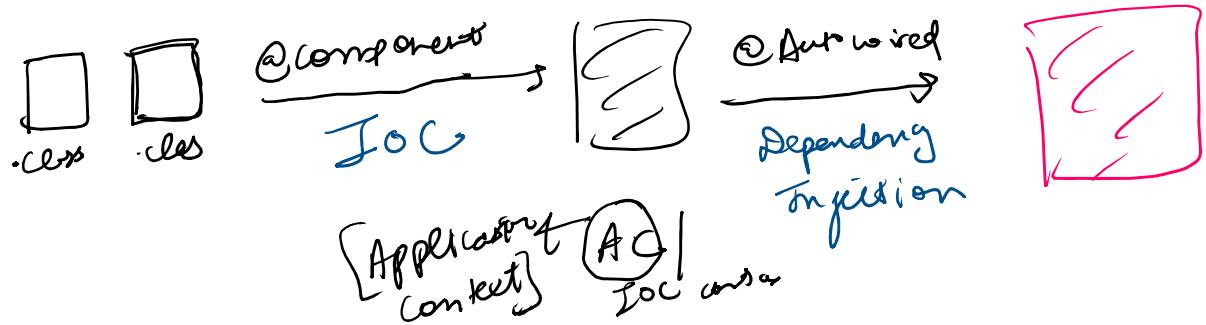
⇒ IOC Container : It's the Context of a Spring application.
Application Context =
responsible for instantiating, creating and managing of Spring beans.



@Auto-wired

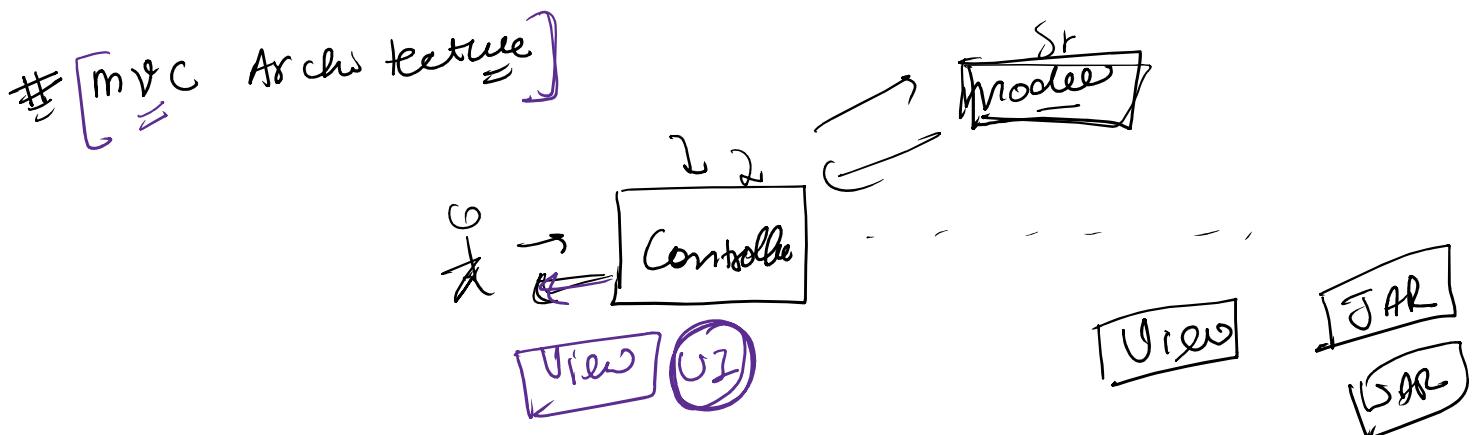
helps to use the beans / components created by Spring

Dependency Injection : Using or injecting the beans created by Spring. @ Auto-wired =



⇒ IoC is based on **Factory Design Pattern**

→ what else we can do with Lombok we can do without lombok as well:



HTTP Status Codes

HTTP Response ⇒ **Response + Status Code**

400 - 499

{ 4XX - Client side error }

500 - 599

{ Server error }

200 - 299 [2XX]

200 → OK

[201 → Created]

500-519
[5xx server side errors].

Response Entity
[helps to send both response + status code]

→ Response Entity is always returned from the controller layer.