**Eager And Lazy Loading**

• As we know Spring loads all singleton beans eagerly in default.

• Prototype beans are loaded only when they are asked for from the context.

• **default-lazy-init** attribute of **</beans>** is used to declare that all

singletons should be loaded lazily by giving **true** value in XML file.

• Or **lazy-init** attribute **of </bean>** can be used to do the same thing per bean basis.

Spring singleton beanleri defaultta **eager** olarak belirtiyor.

Bu prototype beanler icin gecerli değildir. Bunlar da default olarak **lazy** loadingdir.

Singleton icinde lazy load etmek icin ise bir beans icerisinde bütün singletonların lazy olması icin **default-lazy-init** **attribute true value veriyorduk.** Beane özel icin ise Or **lazy-init veriliyor.**

• During bootstrap the container scans all given XML files and create instances of all beans declared as eager.

• If a bean is declared prototype it is not created until asked for from the context even though it is declared to be eagerly loaded.

• Injecting a bean into another bean also causes creation of the bean instance.

• Declaring all singleton beans to be loaded eagerly may cause the bootstrap of the application to take some time but all bean instances would be ready to use when the application finishes starting.

Eğer beanler lazy load edecek sekilde configure edilmisse bu durumda contexten isteninceye kadar bu beanler olusturulmuyor.

Bir beanin yüklenmesine sebep olan davranıs sadece contexten istenince kadar değildir, baska bir beanin inject edilme durumu da buna bir sebeptir.

Prototype beanleri eager load etmeye calıssak bile eager load etmez. Cünkü kac tane bean olusacağını bilmez.