**Lifecycle Callback Methods: Methods**

**Life-Cycle Events in XML – I**

• Lifecycle events can be configured using attributes of **</bean>** in XML:

• **init-method:** For initialization event

• **destroy-method:** For destroy event

• Events for all beans can be configured using attribute of **</beans>** :

**• default-init-method and default-destroy-method**

• These methods are only in affect when they are not overridden in **</bean>.**

XML kullanıyorsak her bean tanımında init-method ve destroy-method attirbuteları var bunlar kullanılabilir.

Bütün beanler icin aynı metot uygulansın istiyorsak **</beans> kullanırız. Bunun üzerindeki** attributelar kullanılabilir.

• Methods given for these attributes don’t receive any paramater but can

have a return type.

• These methods can have any access modifier.

Bu metotlar herhangi bir return type sahip olabilir. Ama hicbir parametre almazlar. Access modifier sahip olabilirler.

**PostConstruct & PreDestroy**

• **PostConstruct** and **PreDestroy** in **java.annotation package**

are two call back method annotations for lifecycle events of a bean.

• They are part of **JSR-250.**

• They can be applied to any method on the bean.

• Methods can be **private**, don’t take any parameter and can return values.

• Multiple lifecycle events can be configured for a bean.

Aynı seyleri postConstruct ve preDestroy ile de yapabilirsin.

Metot isimleri rastgele alabilirler, private olurlar, parametre almazlar, return değeri dönderebilirler. Cok kullanıslı değiller.

JSR-250’nin bir parcası.

Bu anlamda **PostConstruct & PreDestroy kullanmak en mantıklısı.**

**• In order for the container to process these annotations either**

**<context:annotation-config /> or <context:componentscan**

**/> must be in XML file or @ComponentScan must be used.**

**Closable and AutoClosable**

• For beans that **implement java.lang.AutoCloseable** or

**java.io.Closeable interfaces, close()** method can be defined in

XML for pre-destruction event.

**Exception**

• IoC throws

**org.springframework.beans.factory.support.BeanDefinitionValidationException** when it can’t find declared methods in

**</bean>** in XML on bean.

• No exception is thrown if a default notification method declared in

**</beans>** is not found on a bean.