<https://www.dineshonjava.com/application-context-in-spring/>

==================SPRING BEAN LIFE CYCLE=================

1. Spring instantiates the bean.
2. Spring injects values and bean references into the bean’s properties.
3. If the bean implements BeanNameAware, Spring passes the bean’s ID to the set-BeanName() method.
4. If the bean implements BeanFactoryAware, Spring calls the setBeanFactory()

method, passing in the bean factory itself.

1. If the bean implements ApplicationContextAware, Spring will call the set-

ApplicationContext() method, passing in a reference to the enclosing application context.

1. If any of the beans implement the BeanPostProcessor interface, Spring calls

their postProcessBeforeInitialization() method.

1. If any beans implement the InitializingBean interface, Spring calls their

afterPropertiesSet() method. Similarly, if the bean was declared with an

init-method, then the specified initialization method will be called.

1. If there are any beans that implement BeanPostProcessor, Spring will call their postProcessAfterInitialization() method.
2. At this point, the bean is ready to be used by the application and will remain in

the application context until the application context is destroyed.

1. If any beans implement the DisposableBean interface, then Spring will call

their destroy() methods. Likewise, if any bean was declared with a destroymethod, then the specified method will be called.



The org.springframework.beans.factory.xml.XmlBeanFactory is one of the commonly used implementation class for BeanFactory.

The ClassPathXmlApplicationContext , FileSystemXmlApplicationContext and XmlWebApplicationContextare some of the commonly used implementation classes of ApplicationContext interface.

BeanFactory is preferred over ApplicationContext only in scenario’s where resources constraints exist (example mobile apps).