

spring

- [bean 1111](#)
- [bean 111111](#)
- [bean 111111](#)
- [11](#)
- [spring 111111](#)
- [spring 111](#)
- [@SpringBootApplication 1111](#)
- [Bean 1111111111](#)
- [Spring Aop 1 AspectJ 11111](#)
- [Spring 111111](#)
- [ApplicationContext 1 BeanFactory 11](#)
- [springboot 1 spring.factories 11](#)
- [spring 1111](#)

bean ?????

☐☐ bean ☐☐☐

bean []

bean ????????

?????????

1. [] [] **InstantiationAwareBeanPostProcessor** []

```

[ ] bean[ ] BeanPostProcessor[ ] bean
[ ] bean[ ] bean[ ]
[ ] null[ ] bean[ ]
BeanPostProcessors[ ] postProcessAfterInitialization[ ]

```

2. []

3. [] [] **InstantiationAwareBeanPostProcessor** [] bean

```

[ ] Spring
[ ]
[ ] Spring[ ] beaninstance[ ]

```

4. []

5. [] [] **BeanPostProcessor** []

6. []

7. [] [] **BeanPostProcessor** []

8. []

9. []

10. []

11. * 1 []

* 2 []

* 3 []

* 4 [] Aware []

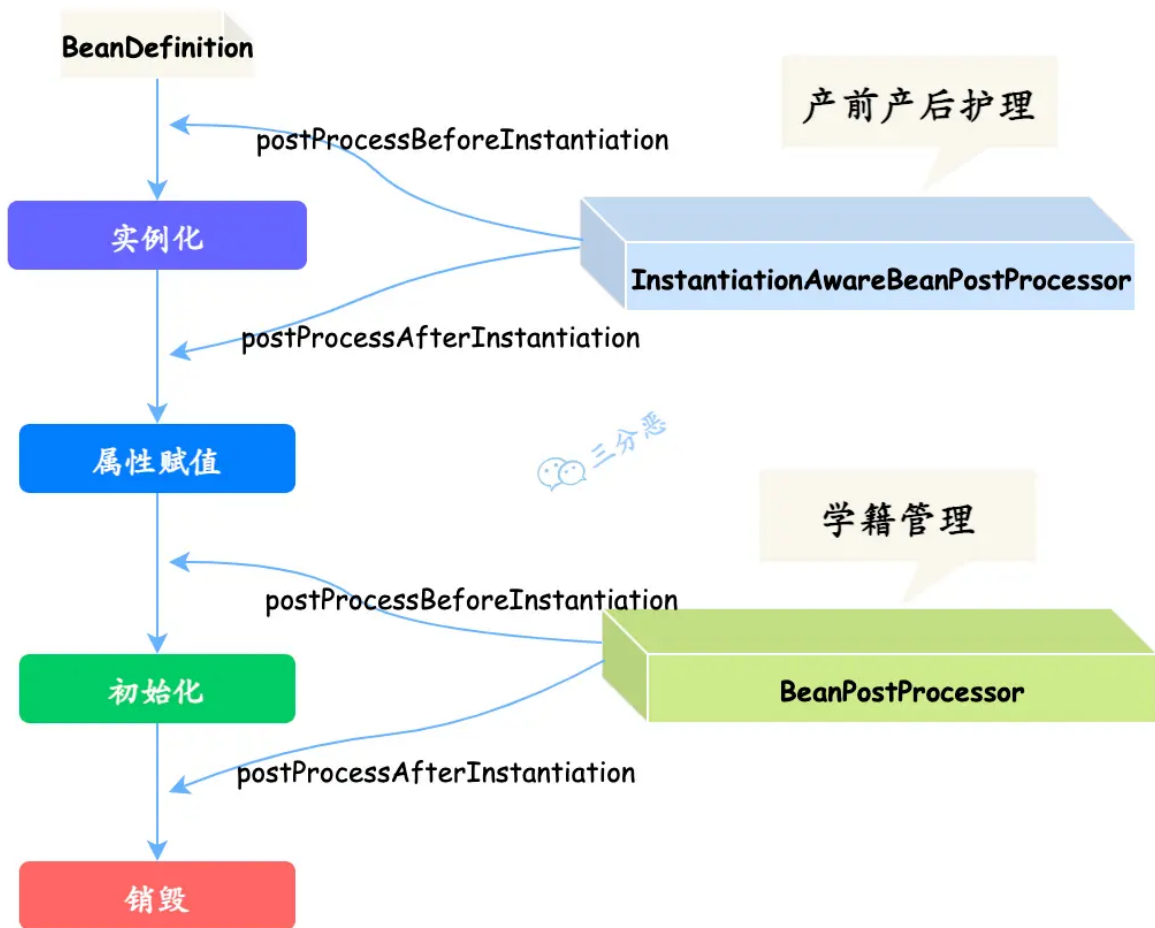
* 5 [] **@PostConstruct**

* 6 [] **InitializingBean**

* 7 [] **AOP** []

* 8 []

???????



ResourcePatternUtils.getResourcePatternResolver

```
    *   2 CachingMetadataReaderFactory
    *   3 "META-INF/spring.components"
    *   noindex null CandidateComponentsIndexLoader.loadIndex
    *   4 register refresh
    *   1 AnnotatedBeanDefinitionReader this.reader.register(componentClasses);
    *   1 doRegisterBean
    *   1 AnnotatedGenericBeanDefinition abd
    *   2 @Conditional
    *   ConfigurationPhase
    *   @Configuration ConfigurationPhase.PARSE_Configuration
    *   3 bean
    *   bean
    *   4 Bean ScopeMetadata
    *   BeanDefinition.attributeNames
    *   this.scopeMetadataResolver.resolveScopeMetadata(abd)
    *   5 bean
    *   singleton bean
    *   bean bean
    *   " singleton
    *   6 processCommonDefinitionAnnotations
    *   7 qualifiers
    *   8 Primary.class == qualifier
    *   9 lazy.class == qualifier
    *   3
    *   bean
    *   AutowireCandidateQualifier
    *   8 Bean BeanDefinitionHolder
    *   9 bean targetbean
    *   "targetBeanName"
    *   ScopedProxyUtils.createScopedProxy
    *   10 bean bean
    *   BeanDefinitionReaderUtils.registerBeanDefinition
    *   5 refresh
    *   1 prepareRefresh()
    *   1
    *   2 initPropertySources()
```

```

00 *      03ConfigurationPropertyResolver#setRequiredProperties
00 *          getEnvironment().validateRequiredProperties()
00 *      04ApplicationListeners
00 05ApplicationEvents
00  this.earlyApplicationEvents = new LinkedHashSet<>()
00 *
00 *      02beanObtainFreshBeanFactory()
00 *      01idBeanFactoryidBeanFactory
00  refreshBeanFactory();
00 *      02ConfigurableListableBeanFactory.getBeanFactory
00 *
00 *      03beanprepareBeanFactory(beanFactory)
00 *      01bean
00  2@code spring.spel.ignore
00  springSpELSpEL<p>“false”
00 *      03PropertyEditorRegistrarbeanPropertyEditor
00  bean
00  registerCustomEditor
00  4beanbeanFactory.addBeanPostProcessor(new
ApplicationContextAwareProcessor(this));
00 *      05
00  BeanFactoryBeanFactoryApplicationContextApplicationContextAware
00  BeanFactoryAware
00  6BeanFactoryMessageSourcebean
00  7beanApplicationListeners
00  beanFactory.addBeanPostProcessor(new ApplicationListenerDetector(this));
00 *      08LoadTimeWeaver
00  9bean
00 *      04beanpostProcessBeanFactory(beanFactory)
00 *
00 *      05beaninvokeBeanFactoryPostProcessors(beanFactory)
00 *      01beanPostProcessorRegistrationDelegate.invokeBeanFactoryPostProcessors
00 *      01PriorityOrdered
00  getBeanApplicationContext
00  PR
00  PostProcessorRegistrationDelegatePR
00  BeanDefinitionRegistryPostProcessors

```


??

spring

spring事务失效的12种场景

访问权限问题

方法用final修饰

方法内部调用

未被spring管理

多线程调用

表不支持事务

错误的传播特性

自己吞了异常

手动抛了别的异常

自定义了回滚异常

嵌套事务回滚多了

spring BeanName BeanName
Bean BeanName Bean BeanName

6.AspectJ SpringAop

SpringAop Bean jdk
Bean CGLIB
SpringAop

AOP AspectJ
AspectJ AspectJ
SpringAop Aop @before @after @Around

7.Spring

spring

REQUIRED

REQUIRES_NEW

SUPPORTS

NOT_SUPPORTED

MANDATORY

NEVER

NESTED

9.BeanFactory ApplicationContext

BeanFactory spring Bean Bean Bean
ApplicationContext
BeanFactory

10.

@Autowrite#####

springboot[] springboot#####

1. Mybatis [] SqlSessionFactory

2. AOP[] EnableAspectJAutoProxy[] []

spring#####

11. Spring#####

1. ##### Spring####

2. ##### BeanDefinition##### Map[]

3. ##### BeanDefinition#### Bean,#### Bean
Bean##### BeanDefiniton[]

4. [] BeanDefinition[] Bean#### Bean#####

5. [] Bean##### spring#####
#####

6. spring#####

12. SpringBoot#####

1. ##### application.properties[] application.yml#####

2. [] Spring#### ApplicationContext##### Spring##### Bean
####

3. ##### springboot##### spring.factories
#####

4. ##### Tomcat,jetty#####

5. ##### springboot#### http##### Contrller,#####

6. ##### springboot#####

13. ##### SpringBoot#####

springboot#####

#####

SB

#####

SB#####

SB classpath

Bean Spring Boot Bean

SB SB

jdk SPI springboot CSDN

14. SpringBoot Starter
1. starter Starter

2. SpringBootStarter

3. Starter

starter

5.15 spring.factories
spring
spring

16 springaop cglib
CGLIB

CGLIB Bean Spring Boot

CGLIB Spring Boot

17. @Bean
spring @Bean bean
BeanDefinition BeanDefinition Bean Bean

18. SpringBoot
spring Bean Bean

BeanFactory Bean getBean Bean

factoryBean getObject

SpringAop jdk cglib

redisTemplate jdbcTemplate

ApplicationListener

Bean

Wrapper BeanWrapper Bean

MVC HandleAdapter

19.

① java servlet ② servlet action ③ action ④ action ⑤ action ⑥ IOC service bean

20.Static

static spring

21.Cookie Session

value Cookie key-

session servlet session currenthashmap http session session

http

22.Component Bean

@Component @Bean Bean

@Component Spring @Bean Bean spring

@Component Spring @Baen Bean

23. 23. 23. url url
DNS DNS URL IP
DNS DNS
IP

TCP syn syn-ack ack

http http

TCP fin ack fin ack

24.

25. Springloc

IOC xml bean BeanDefinition
BeanDefinition IOC BeanDefinition Bean map
ioc

Bean Bean

Bean

Bean ????????????

1. Spring `Bean` `Java`
`Java` `Bean`
2. `Bean` `InitializingBean` `Bean`
`afterPropertiesSet` `init-method`

Spring Aop ?AspectJ?????

1. AspectJ []
2. AOP []
3. [] AspectJ []
4. [] JDK [] CgLib []

Spring ????????

1. `@Configuration`
2. `ApplicationContext`
3. `SpringContext` `SpringContext` `CglibProxy` `cglib`
`ApplicationContext`
`SpringContext`
4. `ApplicationContext`
5. `ApplicationContext`
6. `ApplicationContext` `SpringContext`
7. `ApplicationContext`
8. `final`

ApplicationContext ?

BeanFactory ??

- * 1 `ApplicationContext` `BeanFactory`
- * 2 `ApplicationContext` `BeanFactory`
- * 3 `ApplicationContext`

springboot ?spring.factories ??

1. spring.factories SpringBoot SPI SPI spring.factories
 SpringBoot
2. spring.factories springboot


```

        this.message = message;
    }

    public String getMessage() {
        return message;
    }
}

// 自定义事件
@Component
public class CustomEventListener implements ApplicationListener<CustomEvent> {
    @Override
    public void onApplicationEvent(CustomEvent event) {
        System.out.println("Received CustomEvent: " + event.getMessage());
    }
}

// 自定义事件监听器
@Component
public class AnotherCustomEventListener {
    @EventListener
    public void handleCustomEvent(CustomEvent event) {
        System.out.println("Another CustomEvent received: " + event.getMessage());
    }
}

// 自定义事件发布者
@Component
public class CustomEventPublisher {
    @Autowired
    private ApplicationEventPublisher publisher;

    public void publishCustomEvent(String message) {
        CustomEvent customEvent = new CustomEvent(this, message);
        publisher.publishEvent(customEvent);
    }
}

```

