IOC

###### ioc初始化流程

###### Utils：

ObjectUtils ResourceUtils ClassUtils BeanUtils ReflectionUtils StringUtils

###### **重要类:**

DefaultListableBeanFactory

ApplicationContext AbstractApplicationContext

BeanFactory AbstractBeanFactory

BeanDefinition AbstractBeanDefinition

RootBeanDefinition

AnnotationConfigApplicationContext

###### **三级缓存：解决循环依赖**

###### 不解决构造方法注入的循环依赖，不解决非单例的循环引用

singletonObjects 已创建单例 创建完成加入

earlySingletonObjects 早期对象 getSingleton() 创建完成删除

singletonFactories 创建对象的工厂对象 获取早期对象

单例bean的创建策略

private InstantiationStrategy instantiationStrategy = new CglibSubclassingInstantiationStrategy();

AOP:

AspectJAutoProxyBeanDefinitionParser.parse 注册

AnnotationAwareAspectJAutoProxyCreator

此bean org.springframework.aop.config.internalAutoProxyCreator 初始化 aspectJAdvisorFactory

tomcatServletWebServerFactory

InstantiationAwareBeanPostProcessor先执行：

postProcessBeforeInstantiation(Class<?> beanClass, String beanName)

postProcessAfterInstantiation(Object bean, String beanName)

BeanPostProcessor再执行：

postProcessBeforeInitialization(Object bean, String beanName)

Object postProcessAfterInitialization(Object bean, String beanName)

AOP的入口是AbstractAutowireCapableBeanFactory.doCreateBean()方法。

1.匹配切点方法（构建切入点表达式类和切面类）

2.创建代理对象

###### MVC

MVC 的入口是DispatcherServlet.doDispatch()方法。

事务

PlatformTransactionManager

TransactionDefinition

TransactionStatus