一、三种注入方式

1、有参构造器（有循环依赖的问题）

2、Set

3、注解（先通过无参构造器创建实例，通过反射填充属性）

二、循环依赖

1、有参构造器注入的没法解决

2、<https://blog.csdn.net/chejinqiang/article/details/80003868>

<http://www.imooc.com/article/34150>



三、单例里获取多例

1、在bean A中引入ApplicationContext每次调用方法时用上下文的getBean(name,class)方法去重新获取bean B的实例。



2、使用@Lookup注解。

3、<https://somefuture.iteye.com/blog/2404846>

1. @Scope(value = ConfigurableBeanFactory.SCOPE\_PROTOTYPE, proxyMode = ScopedProxyMode.TARGET\_CLASS)

四、controller层的request使用@autowired没有线程安全问题

<https://blog.csdn.net/forlovedoit/article/details/53204667>

我们在Controller层中所注入的HttpServletRequest其实是一个JDK动态代理生成的对象

五、JDK动态代理和CGLIB字节码生成的区别

JDK动态代理只能对实现了接口的类生成代理，而不能针对类。

CGLIB是针对类实现代理，主要是对指定的类生成一个子类，覆盖其中的方法，并覆盖其中方法实现增强，但是因为采用的是继承，所以该类或方法最好不要声明成final，对于final类或方法，是无法继承的

SpringMVC源码：

<https://blog.csdn.net/gududedabai/article/details/83352106>

<https://blog.csdn.net/gududedabai/article/details/83375156>

< url-pattern > / </ url-pattern > 不会匹配到\*.jsp，即：\*.jsp不会进入spring的 DispatcherServlet类 。

< url-pattern > /\* </ url-pattern > 会匹配\*.jsp，会出现返回jsp视图时再次进入spring的