1. spring的bean工厂会扫描指定路径，将类按顺序进行初始化，并在初始化时就建立代理，然后放入map中（问题：暂不知道以什么基准来排序，根据自己的做法看应该是扫描出来的文件顺序？）
2. 建立实例之后，对成员变量进行注入，在注入过程中，会通过getBean获得对应类型（名称）的bean，若bean已经存在，则直接注入，若不存在，则参照1进行实例化，建立代理，然后将代理实例注入
3. spring的beanFactory会在初始化bean的时候建立依赖链（结构使用了HashSet）
4. springboot 中的spring.aop.proxy-target-class默认为true，即若不手动设置为false，会默认使用cglib建立代理类
5. spring中建立代理的方式其实很简单粗暴，（在spring.aop.proxy-target-class设置为false的情况下）若检查到代理的目标对象targetBean有父接口，则使用jdk方式建立动态代理；若targetBean没有实现接口，则使用cglib方式建立代理。
6. 使用Annotation类型的对象getClass，会错误的获取一个Proxy的class对象（暂不清楚原因，需要使用annotationType才能正确获得想要的注解的class对象
7. 使用cglib创建多重代理，会导致newInstance方法重复建立而导致

Caused by: java.lang.ClassFormatError: Duplicate method name "newInstance" with signature "([Lnet.sf.cglib.proxy.Callback;)Ljava.lang.Object;" in class file com/borris/smt/controller/TestController$$EnhancerByCGLIB$$9004d5fa$$EnhancerByCGLIB$$dab4f5dd

以上异常，因此需要构建一个可以递归执行的方法栈来达到多重代理的效果。