源自：<http://www.iteye.com/topic/1122937>

| **锁定老帖子**[**主题：Spring开闭原则的表现-BeanPostProcessor扩展点-2**](http://www.iteye.com/topic/1122937)  **该帖已经被评为精华帖** | | | |
| --- | --- | --- | --- |
| **作者** | | **正文** | |
| * **jinnianshilongnian** * 等级: 五钻会员 * [jinnianshilongnian的博客](http://jinnianshilongnian.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 1112 * 积分: 2280 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-20   最后修改：2012-04-20  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=Spring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937)  [<](http://www.iteye.com/topic/1122937) [>](http://www.iteye.com/topic/1122937) 猎头职位: [**上海:**Junior Product Manager](http://www.iteye.com/jobs/2509)  相关文章:    * [Spring开闭原则的表现-BeanPostProcessor扩展点-1](http://www.iteye.com/topic/1122859) * [Spring事务处理时自我调用的解决方案及一些实现方式的风险](http://www.iteye.com/topic/1122740) * [SpringAOP嵌套调用的解决办法](http://www.iteye.com/topic/109236)   推荐群组: [springMVC](http://springmvc.group.iteye.com/)  [更多相关推荐](http://www.iteye.com/wiki/topic/1122937)  [Spring](http://www.iteye.com/forums/tag/Spring)      上接[**Spring提供的BeanPostProcessor的扩展点-1**](http://www.iteye.com/topic/1122859)继续分析。       四、BeanPostProcessor接口及回调方法图 http://dl.iteye.com/upload/attachment/0066/9379/e2e1a684-ebeb-3e3f-8cd5-21471a239d16.png  从图中我们可以看出一共五个接口，共十个回调方法，即十个扩展点，但我们之前的文章只分析了其中八个，另外两个稍候也会解析一下是干什么的。    ===================================================================  =================================================================== 五、五个接口十个扩展点解析 **1、InstantiationAwareBeanPostProcessor**：实例化Bean后置处理器（继承BeanPostProcessor）  postProcessBeforeInstantiation ：在实例化目标对象之前执行，可以自定义实例化逻辑，如返回一个代理对象等，（3.1处执行；如果此处返回的Bean不为null将中断后续Spring创建Bean的流程，且只执行postProcessAfterInitialization回调方法，如当AbstractAutoProxyCreator的实现者注册了TargetSourceCreator（创建自定义的TargetSource）将改变执行流程，不注册TargetSourceCreator我们默认使用的是SingletonTargetSource（即AOP代理直接保证目标对象），此处我们还可以使用如ThreadLocalTargetSource（线程绑定的Bean）、CommonsPoolTargetSource（实例池的Bean）等等，大家可以去spring官方文档了解TargetSource详情；  postProcessAfterInitialization ： Bean实例化完毕后执行的后处理操作，所有初始化逻辑、装配逻辑之前执行，如果返回false将阻止其他的InstantiationAwareBeanPostProcessor的postProcessAfterInstantiation的执行，（3.2和（9.1处执行；在此处可以执行一些初始化逻辑或依赖装配逻辑；  postProcessPropertyValues ：完成其他定制的一些依赖注入和依赖检查等，如AutowiredAnnotationBeanPostProcessor执行@Autowired注解注入，CommonAnnotationBeanPostProcessor执行@Resource等注解的注入，PersistenceAnnotationBeanPostProcessor执行@ PersistenceContext等JPA注解的注入，RequiredAnnotationBeanPostProcessor执行@ Required注解的检查等等，（9.3处执行；    **2、MergedBeanDefinitionPostProcessor**：合并Bean定义后置处理器         （继承BeanPostProcessor）  postProcessMergedBeanDefinition：执行Bean定义的合并，在（7.1处执行，且在实例化完Bean之后执行；    **3、SmartInstantiationAwareBeanPostProcessor**：智能实例化Bean后置处理器（继承InstantiationAwareBeanPostProcessor）  predictBeanType：预测Bean的类型，返回第一个预测成功的Class类型，如果不能预测返回null；当你调用BeanFactory.getType(name)时当通过Bean定义无法得到Bean类型信息时就调用该回调方法来决定类型信息；BeanFactory.isTypeMatch(name, targetType)用于检测给定名字的Bean是否匹配目标类型（如在依赖注入时需要使用）；  determineCandidateConstructors：检测Bean的构造器，可以检测出多个候选构造器，再有相应的策略决定使用哪一个，如AutowiredAnnotationBeanPostProcessor实现将自动扫描通过@Autowired/@Value注解的构造器从而可以完成构造器注入，请参考[【第十二章】零配置 之 12.2 注解实现Bean依赖注入 ——跟我学spring3](http://jinnianshilongnian.iteye.com/blog/1457224) ，（6.2.2.1处执行；  getEarlyBeanReference：当正在创建A时，A依赖B，此时通过（8将A作为ObjectFactory放入单例工厂中进行early expose，此处B需要引用A，但A正在创建，从单例工厂拿到ObjectFactory（其通过getEarlyBeanReference获取及早暴露Bean），从而允许循环依赖，此时AspectJAwareAdvisorAutoProxyCreator（完成xml风格的AOP配置(<aop:config>)将目标对象（A）包装到AOP代理对象）或AnnotationAwareAspectJAutoProxyCreator（完成@Aspectj注解风格（<aop:aspectj-autoproxy> @Aspect）将目标对象（A）包装到AOP代理对象），其返回值将替代原始的Bean对象，即此时通过early reference能得到正确的代理对象，（8.1处实施；如果此处执行了，（10.3.3处的AspectJAwareAdvisorAutoProxyCreator或AnnotationAwareAspectJAutoProxyCreator的postProcessAfterInitialization将不执行，即这两个回调方法是二选一的；    **4、BeanPostProcessor**：Bean后置处理器  postProcessBeforeInitialization：实例化、依赖注入完毕，在调用显示的初始化之前完成一些定制的初始化任务，如BeanValidationPostProcessor完成JSR-303 @Valid注解Bean验证，InitDestroyAnnotationBeanPostProcessor完成@PostConstruct注解的初始化方法调用，ApplicationContextAwareProcessor完成一些Aware接口的注入（如EnvironmentAware、ResourceLoaderAware、ApplicationContextAware），其返回值将替代原始的Bean对象；（10.2处执行；  postProcessAfterInitialization：实例化、依赖注入、初始化完毕时执行，如AspectJAwareAdvisorAutoProxyCreator（完成xml风格的AOP配置(<aop:config>)的目标对象包装到AOP代理对象）、AnnotationAwareAspectJAutoProxyCreator（完成@Aspectj注解风格（<aop:aspectj-autoproxy> @Aspect）的AOP配置的目标对象包装到AOP代理对象），其返回值将替代原始的Bean对象；（10.3.3处执行；此处需要参考getEarlyBeanReference；    **5、DestructionAwareBeanPostProcessor**：销毁Bean后置处理器（继承BeanPostProcessor）  postProcessBeforeDestruction：销毁后处理回调方法，该回调只能应用到单例Bean，如InitDestroyAnnotationBeanPostProcessor完成@PreDestroy注解的销毁方法调用；（12.1.1处执行。    ===================================================================  =================================================================== 六、内置的一些BeanPostProcessor http://dl.iteye.com/upload/attachment/0066/9383/1590f6b2-4ba4-3d81-b103-6d20a9a04012.png   部分内置的BeanPostProcessor  此图只有内置的一部分。   1、ApplicationContextAwareProcessor 容器启动时会自动注册。注入那些实现ApplicationContextAware、MessageSourceAware、ResourceLoaderAware、EnvironmentAware、  EmbeddedValueResolverAware、ApplicationEventPublisherAware标识接口的Bean需要的相应实例，在postProcessBeforeInitialization回调方法中进行实施，即（10.2处实施。   2、CommonAnnotationBeanPostProcessor CommonAnnotationBeanPostProcessor继承InitDestroyAnnotationBeanPostProcessor，当在配置文件有<context:annotation-config>或<context:component-scan>会自动注册。    提供对JSR-250规范注解的支持@javax.annotation.Resource、@javax.annotation.PostConstruct和@javax.annotation.PreDestroy等的支持。    2.1、通过@Resource注解进行依赖注入：      postProcessPropertyValues：通过此回调进行@Resource注解的依赖注入；（9.3处实施；  2.2、用于执行@PostConstruct 和@PreDestroy 注解的初始化和销毁方法的扩展点：      postProcessBeforeInitialization()将会调用bean的@PostConstruct方法；（10.2处实施；      postProcessBeforeDestruction()将会调用单例 Bean的@PreDestroy方法（此回调方法会在容器销毁时调用），（12.1.1处实施。    详见[【第十二章】零配置 之 12.2 注解实现Bean依赖注入 ——跟我学spring3](http://jinnianshilongnian.iteye.com/blog/1457224)，JSR-250注解部分。 3、AutowiredAnnotationBeanPostProcessor 当在配置文件有<context:annotation-config>或<context:component-scan>会自动注册。    提供对JSR-330规范注解的支持和Spring自带注解的支持。    3.1、Spring自带注解的依赖注入支持，@Autowired和@Value：      determineCandidateConstructors ：决定候选构造器；详见【12.2中的构造器注入】；（6.2.2.1处实施；  postProcessPropertyValues ：进行依赖注入；详见【12.2中的字段注入和方法参数注入】；（9.3处实施；  3.2、对JSR-330规范注解的依赖注入支持，@Inject：      同2.1类似只是查找使用的注解不一样；    详见[【第十二章】零配置 之 12.2 注解实现Bean依赖注入 ——跟我学spring3](http://jinnianshilongnian.iteye.com/blog/1457224)，Spring自带依赖注入注解和 JSR-330注解部分。 4、RequiredAnnotationBeanPostProcessor 当在配置文件有<context:annotation-config>或<context:component-scan>会自动注册。    4.1、提供对@ Required注解的方法进行依赖检查支持：      postProcessPropertyValues：如果检测到没有进行依赖注入时抛出BeanInitializationException异常；（9.3处实施；    详见[【第十二章】零配置 之 12.2 注解实现Bean依赖注入 ——跟我学spring3](http://jinnianshilongnian.iteye.com/blog/1457224)，**@Required：依赖检查**。 5、PersistenceAnnotationBeanPostProcessor 当在配置文件有<context:annotation-config>或<context:component-scan>会自动注册。    5.1、通过对JPA @ javax.persistence.PersistenceUnit和@ javax.persistence.PersistenceContext注解进行依赖注入的支持；      postProcessPropertyValues ： 根据@PersistenceUnit/@PersistenceContext进行EntityManagerFactory和EntityManager的支持； 6、AbstractAutoProxyCreator AspectJAwareAdvisorAutoProxyCreator和AnnotationAwareAspectJAutoProxyCreator都是继承AbstractAutoProxyCreator，AspectJAwareAdvisorAutoProxyCreator提供对（<aop:config>）声明式AOP的支持，AnnotationAwareAspectJAutoProxyCreator提供对（<aop:aspectj-autoproxy>）注解式（@AspectJ）AOP的支持，因此只需要分析AbstractAutoProxyCreator即可。    当使用<aop:config>配置时自动注册AspectJAwareAdvisorAutoProxyCreator，而使用<aop:aspectj-autoproxy>时会自动注册AnnotationAwareAspectJAutoProxyCreator。    6.1、predictBeanType：预测Bean的类型，如果目标对象被AOP代理对象包装，此处将返回AOP代理对象的类型；  **Java代码  [收藏代码](javascript:void())**   1. **public** Class<?> predictBeanType(Class<?> beanClass, String beanName) { 2. Object cacheKey = getCacheKey(beanClass, beanName); 3. **return** **this**.proxyTypes.get(cacheKey); //获取代理对象类型，可能返回null 4. }   6.2、postProcessBeforeInstantiation：  **Java代码  [收藏代码](javascript:void())**   1. **public** Object postProcessBeforeInstantiation(Class<?> beanClass, String beanName) **throws** BeansException { 2. //1、得到一个缓存的唯一key（根据beanClass和beanName生成唯一key） 3. Object cacheKey = getCacheKey(beanClass, beanName); 4. //2、如果当前targetSourcedBeans（通过自定义TargetSourceCreator创建的TargetSource）不包含cacheKey 5. **if** (!**this**.targetSourcedBeans.contains(cacheKey)) { 6. //2.1、advisedBeans（已经被增强的Bean，即AOP代理对象）中包含当前cacheKey或nonAdvisedBeans（不应该被增强的Bean）中包含当前cacheKey 返回null，即走Spring默认流程 7. **if** (**this**.advisedBeans.contains(cacheKey) || **this**.nonAdvisedBeans.contains(cacheKey)) { 8. **return** **null**; 9. } 10. //2.2、如果是基础设施类（如Advisor、Advice、AopInfrastructureBean的实现）不进行处理 11. //2.2、shouldSkip 默认false，可以生成子类覆盖，如AspectJAwareAdvisorAutoProxyCreator覆盖    （if (((AbstractAspectJAdvice) advisor.getAdvice()).getAspectName().equals(beanName)) return true;  即如果是自己就跳过） 12. **if** (isInfrastructureClass(beanClass) || shouldSkip(beanClass, beanName)) { 13. **this**.nonAdvisedBeans.add(cacheKey);//在不能增强的Bean列表缓存当前cacheKey 14. **return** **null**; 15. } 16. } 18. //3、开始创建AOP代理对象 19. //3.1、配置自定义的TargetSourceCreator进行TargetSource创建 20. TargetSource targetSource = getCustomTargetSource(beanClass, beanName); 21. **if** (targetSource != **null**) { 22. //3.2、如果targetSource不为null 添加到targetSourcedBeans缓存，并创建AOP代理对象 23. **this**.targetSourcedBeans.add(beanName); 24. // specificInterceptors即增强（包括前置增强、后置增强等等） 25. Object[] specificInterceptors = getAdvicesAndAdvisorsForBean(beanClass, beanName, targetSource); 26. //3.3、创建代理对象 27. Object proxy = createProxy(beanClass, beanName, specificInterceptors, targetSource); 28. //3.4、将代理类型放入proxyTypes从而允许后续的predictBeanType()调用获取 29. **this**.proxyTypes.put(cacheKey, proxy.getClass()); 30. **return** proxy; 31. } 32. **return** **null**; 33. }     从如上代码可以看出，当我们配置TargetSourceCreator进行自定义TargetSource创建时，会创建代理对象并中断默认Spring创建流程。    6.3、getEarlyBeanReference  **Java代码  [收藏代码](javascript:void())**   1. //获取early Bean引用（只有单例Bean才能回调该方法） 2. **public** Object getEarlyBeanReference(Object bean, String beanName) **throws** BeansException { 3. Object cacheKey = getCacheKey(bean.getClass(), beanName); 4. //1、将cacheKey添加到earlyProxyReferences缓存，从而避免多次重复创建 5. **this**.earlyProxyReferences.add(cacheKey); 6. //2、包装目标对象到AOP代理对象（如果需要） 7. **return** wrapIfNecessary(bean, beanName, cacheKey); 8. }     6.4、postProcessAfterInitialization  **Java代码  [收藏代码](javascript:void())**   1. **public** Object postProcessAfterInitialization(Object bean, String beanName) **throws** BeansException { 2. **if** (bean != **null**) { 3. Object cacheKey = getCacheKey(bean.getClass(), beanName); 4. //1、如果之前调用过getEarlyBeanReference获取包装目标对象到AOP代理对象（如果需要），则不再执行 5. **if** (!**this**.earlyProxyReferences.contains(cacheKey)) { 6. //2、包装目标对象到AOP代理对象（如果需要） 7. **return** wrapIfNecessary(bean, beanName, cacheKey); 8. } 9. } 10. **return** bean; 11. }     **Java代码  [收藏代码](javascript:void())**   1. **protected** Object wrapIfNecessary(Object bean, String beanName, Object cacheKey) { 2. **if** (**this**.targetSourcedBeans.contains(beanName)) {//通过TargetSourceCreator进行自定义TargetSource不需要包装 3. **return** bean; 4. } 5. **if** (**this**.nonAdvisedBeans.contains(cacheKey)) {//不应该被增强对象不需要包装 6. **return** bean; 7. } 8. **if** (isInfrastructureClass(bean.getClass()) || shouldSkip(bean.getClass(), beanName)) {//基础设施/应该skip的不需要保证 9. **this**.nonAdvisedBeans.add(cacheKey); 10. **return** bean; 11. } 13. // 如果有增强就执行包装目标对象到代理对象 14. Object[] specificInterceptors = getAdvicesAndAdvisorsForBean(bean.getClass(), beanName, **null**); 15. **if** (specificInterceptors != DO\_NOT\_PROXY) { 16. **this**.advisedBeans.add(cacheKey);//将cacheKey添加到已经被增强列表，防止多次增强 17. Object proxy = createProxy(bean.getClass(), beanName, specificInterceptors, **new** SingletonTargetSource(bean));//创建代理对象 18. **this**.proxyTypes.put(cacheKey, proxy.getClass());//缓存代理类型 19. **return** proxy; 20. } 21. **this**.nonAdvisedBeans.add(cacheKey); 22. **return** bean; 23. }     从如上流程可以看出 getEarlyBeanReference和postProcessAfterInitialization是二者选一的，而且单例Bean目标对象只能被增强一次，而原型Bean目标对象可能被包装多次。   7、BeanValidationPostProcessor 默认不自动注册，Spring3.0开始支持。    提供对JSR-303验证规范支持。    根据afterInitialization是false/true决定调用postProcessBeforeInitialization或postProcessAfterInitialization来通过JSR-303规范验证Bean，默认false。 8、MethodValidationPostProcessor Spring3.1开始支持，且只支持Hibernate Validator 4.2及更高版本，从Spring 3.2起可能将采取自动检测Bean Validation 1.1兼容的提供商且自动注册（Bean Validation 1.1 (JSR-349)正处于草案阶段，它将提供方法级别的验证，提供对方法级别的验证），目前默认不自动注册。    Bean Validation 1.1草案请参考<http://jcp.org/en/jsr/detail?id=349>    <http://beanvalidation.org/>。    提供对方法参数/方法返回值的进行验证（即前置条件/后置条件的支持），通过JSR-303注解验证，使用方式如：  public @NotNull Object myValidMethod(@NotNull String arg1, @Max(10) int arg2)    默认只对@org.springframework.validation.annotation.Validated注解的Bean进行验证，我们可以修改validatedAnnotationType为其他注解类型来支持其他注解验证。而且目前只支持Hibernate Validator实现，在未来版本可能支持其他实现。    有了这东西之后我们就不需要在进行如Assert.assertNotNull（）这种前置条件/后置条件的判断了。 9、ScheduledAnnotationBeanPostProcessor 当配置文件中有<task:annotation-driven>自动注册或@EnableScheduling自动注册。    提供对注解@Scheduled任务调度的支持。    postProcessAfterInitialization：通过查找Bean对象类上的@Scheduled注解来创建ScheduledMethodRunnable对象并注册任务调度方法（仅返回值为void且方法是无形式参数的才可以）。    可参考Spring官方文档的任务调度章节学习@Scheduled注解任务调度。 10、AsyncAnnotationBeanPostProcessor 当配置文件中有<task:annotation-driven>自动注册或@EnableAsync自动注册。    提供对@ Async和EJB3.1的@javax.ejb.Asynchronous注解的异步调用支持。    postProcessAfterInitialization：通过ProxyFactory创建目标对象的代理对象，默认使用AsyncAnnotationAdvisor（内部使用AsyncExecutionInterceptor 通过AsyncTaskExecutor（继承TaskExecutor）通过submit提交异步任务）。    可参考Spring官方文档的异步调用章节学习@Async注解异步调用。 11、ServletContextAwareProcessor 在使用Web容器时自动注册。    类似于ApplicationContextAwareProcessor，当你的Bean 实现了ServletContextAware/ ServletConfigAware会自动调用回调方法注入ServletContext/ ServletConfig。    ===================================================================  =================================================================== 七、BeanPostProcessor如何注册 1、如ApplicationContextAwareProcessor会在ApplicationContext容器启动时自动注册，而CommonAnnotationBeanPostProcessor和AutowiredAnnotationBeanPostProcessor会在当你使用<context:annotation-config>或<context:component-scan>配置时自动注册。  2、只要将BeanPostProcessor注册到容器中，Spring会在启动时自动获取并注册。    ===================================================================  =================================================================== 八、BeanPostProcessor的执行顺序 1、如果使用BeanFactory实现，非ApplicationContext实现，BeanPostProcessor执行顺序就是添加顺序。    2、如果使用的是AbstractApplicationContext（实现了ApplicationContext）的实现，则通过如下规则指定顺序。  2.1、PriorityOrdered（继承了Ordered），实现了该接口的BeanPostProcessor会在第一个顺序注册，标识高优先级顺序，即比实现Ordered的具有更高的优先级；  2.2、Ordered，实现了该接口的BeanPostProcessor会第二个顺序注册；    int HIGHEST\_PRECEDENCE = Integer.MIN\_VALUE;//最高优先级  int LOWEST\_PRECEDENCE = Integer.MAX\_VALUE;//最低优先级    即数字越小优先级越高，数字越大优先级越低，如0（高优先级）——1000（低优先级）    2.3、无序的，没有实现Ordered/ PriorityOrdered的会在第三个顺序注册；  2.4、内部Bean后处理器，实现了MergedBeanDefinitionPostProcessor接口的是内部Bean PostProcessor，将在最后且无序注册。      3、接下来我们看看内置的BeanPostProcessor执行顺序    //1注册实现了PriorityOrdered接口的BeanPostProcessor    //2注册实现了Ordered接口的BeanPostProcessor  AbstractAutoProxyCreator              实现了Ordered，order = Ordered.LOWEST\_PRECEDENCE  MethodValidationPostProcessor          实现了Ordered，LOWEST\_PRECEDENCE  ScheduledAnnotationBeanPostProcessor   实现了Ordered，LOWEST\_PRECEDENCE  AsyncAnnotationBeanPostProcessor      实现了Ordered，order = Ordered.LOWEST\_PRECEDENCE    //3注册无实现任何接口的BeanPostProcessor  BeanValidationPostProcessor            无序  ApplicationContextAwareProcessor       无序  ServletContextAwareProcessor          无序    //3 注册实现了MergedBeanDefinitionPostProcessor接口的BeanPostProcessor，且按照实现了Ordered的顺序进行注册，没有实现Ordered的默认为Ordered.LOWEST\_PRECEDENCE。  PersistenceAnnotationBeanPostProcessor  实现了PriorityOrdered，Ordered.LOWEST\_PRECEDENCE - 4  AutowiredAnnotationBeanPostProcessor   实现了PriorityOrdered，order = Ordered.LOWEST\_PRECEDENCE - 2  RequiredAnnotationBeanPostProcessor    实现了PriorityOrdered，order = Ordered.LOWEST\_PRECEDENCE - 1  CommonAnnotationBeanPostProcessor    实现了PriorityOrdered，Ordered.LOWEST\_PRECEDENCE    从上到下顺序执行，如果order相同则我们应该认为同序（谁先执行不确定，其执行顺序根据注册顺序决定）。    ===================================================================  =================================================================== 九、完成Spring事务处理时自我调用的解决方案及一些实现方式的风险分析 场景请先参考请参考[Spring事务处理时自我调用的解决方案及一些实现方式的风险](http://www.iteye.com/topic/1122740)中的3.3、通过BeanPostProcessor 在目标对象中注入代理对象。      **分析：**  http://dl.iteye.com/upload/attachment/0066/9385/334849c0-7186-3cd0-b981-f1d3664aa03a.png      **问题出现在5和9处：**    5、使用步骤1处注册的SingletonFactory（ObjectFactory.getObject() 使用AnnotationAwareAspectJAutoProxyCreator的getEarlyBeanReference获取循环引用Bean），因此此处将返回A目标对象的代理对象；    9、此处调用AnnotationAwareAspectJAutoProxyCreator的postProcessAfterInitialization，但发现之前调用过AnnotationAwareAspectJAutoProxyCreator的getEarlyBeanReference获取代理对象，此处不再创建代理对象，而是直接返回目标对象，因此使用InjectBeanSelfProcessor不能注入代理对象；但此时的Spring容器中的A已经是代理对象了，因此我使用了从上下文重新获取A代理对象的方式注入（context.getBean(beanName)）。    此处的getEarlyBeanReference和postProcessAfterInitialization为什么是二者选一的请参考之前介绍的AbstractAutoProxyCreator。    到此问题我们分析完毕，实际项目中的循环依赖应该尽量避免，这违反了“无环依赖原则”。        下一篇我将介绍一些内置BeanPostProcessor的使用和自定义一些自己的BeanPostProcessor来更好的理解这些扩展点。   * [查看图片附件](http://www.iteye.com/topic/1122937)   声明：ITeye文章版权属于作者，受法律保护。没有作者书面许可不得转载。  推荐链接 | |
| [返回顶楼](http://www.iteye.com/topic/1122937) | |  | |
|  | | | |
| * **ajavamark** * 等级: 初级会员 * [ajavamark的博客](http://ajavamark.iteye.com/) * 文章: 1 * 积分: 30 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-20  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=ajavamark%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%232343357)  好文，多谢分享 | |
| [返回顶楼](http://www.iteye.com/topic/1122937) | | [回帖地址](http://www.iteye.com/topic/1122937#2343357)  [0](http://www.iteye.com/topic/1122937) [0](http://www.iteye.com/topic/1122937) 已投票 | |
|  | | | |
| * **jinnianshilongnian** * 等级: 五钻会员 * [jinnianshilongnian的博客](http://jinnianshilongnian.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 1112 * 积分: 2280 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-21  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=jinnianshilongnian%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%232343406)  **grzrt 写道**  支持学习精神  谢谢兄弟支持，希望一起来探讨，对BeanPostProcessor你有什么观点吗？ | |
| [返回顶楼](http://www.iteye.com/topic/1122937) | | [回帖地址](http://www.iteye.com/topic/1122937#2343406)  [0](http://www.iteye.com/topic/1122937) [0](http://www.iteye.com/topic/1122937) 已投票 | |
|  | | | |
| * **Sunyx** * 等级: 初级会员 * [Sunyx的博客](http://sunyx.iteye.com/) * 文章: 2 * 积分: 30 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-21  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=Sunyx%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%232343432)  虽然不懂，但是还是支持分享的精神 | |
| [返回顶楼](http://www.iteye.com/topic/1122937) | | [回帖地址](http://www.iteye.com/topic/1122937#2343432)  [0](http://www.iteye.com/topic/1122937) [0](http://www.iteye.com/topic/1122937) 已投票 | |
|  | | | |
| * **jinnianshilongnian** * 等级: 五钻会员 * [jinnianshilongnian的博客](http://jinnianshilongnian.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 1112 * 积分: 2280 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-21  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=jinnianshilongnian%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%232343462)  **Sunyx 写道**  虽然不懂，但是还是支持分享的精神  希望能一起探讨，学习是一种信仰，能使我们进步http://www.iteye.com/images/smiles/icon_biggrin.gif http://www.iteye.com/images/smiles/icon_biggrin.gif | |
| [返回顶楼](http://www.iteye.com/topic/1122937) | | [回帖地址](http://www.iteye.com/topic/1122937#2343462)  [0](http://www.iteye.com/topic/1122937) [0](http://www.iteye.com/topic/1122937) 已投票 | |
|  | | | |
| * **caoxudong818** * 等级: 一星会员 * [caoxudong818的博客](http://caoxudong818.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 76 * 积分: 170 * 来自: 北京 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-21  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=caoxudong818%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%232343486)  spring系列很不错 | |
| [返回顶楼](http://www.iteye.com/topic/1122937) | | [回帖地址](http://www.iteye.com/topic/1122937#2343486)  [0](http://www.iteye.com/topic/1122937) [0](http://www.iteye.com/topic/1122937) 已投票 | |
|  | | | |
| * **jinnianshilongnian** * 等级: 五钻会员 * [jinnianshilongnian的博客](http://jinnianshilongnian.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 1112 * 积分: 2280 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-21  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=jinnianshilongnian%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%232343508)  **caoxudong818 写道**  spring系列很不错  欢迎兄弟一起来分享spring相关的技术及一些常见问题 | |
| [返回顶楼](http://www.iteye.com/topic/1122937) | | [回帖地址](http://www.iteye.com/topic/1122937#2343508)  [0](http://www.iteye.com/topic/1122937) [0](http://www.iteye.com/topic/1122937) 已投票 | |
|  | | | |
| * **guofengcn** * 等级: 初级会员 * [guofengcn的博客](http://guofengcn.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 31 * 积分: 30 * 来自: 北京 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-21  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=guofengcn%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%232343536)  问一个问题：   我想扩展MappingJacksonHttpMessageConverter（或者自己实现HttpMessageConverter用别的Json处理lib来实现）在方法上使用注解@ResponseBody 来返回json。   但是遇到一个问题，我们返回的对象，比如User，是有很多属性的，并不需要序列化所有的属性，而且是每个方法上不一样的（所以不能直接在实体上写Jackson的ignore注解来实现），想在controller层的方法上加个自定义注解来实现，但是HttpMessageConverter这个接口，并没有传入任何当前的handlerMethod，所以也就无法获取当前方法上的我自己定义的注解。  不知道有没有什么好的方案来实现这样的功能？http://www.iteye.com/images/smiles/icon_biggrin.gif | |
| [返回顶楼](http://www.iteye.com/topic/1122937) | | [回帖地址](http://www.iteye.com/topic/1122937#2343536)  [0](http://www.iteye.com/topic/1122937) [0](http://www.iteye.com/topic/1122937) 已投票 | |
|  | | | |
| * **guofengcn** * 等级: 初级会员 * [guofengcn的博客](http://guofengcn.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 31 * 积分: 30 * 来自: 北京 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-21   最后修改：2012-04-21  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=guofengcn%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%232343540)  **guofengcn 写道**  问一个问题：   我想扩展MappingJacksonHttpMessageConverter（或者自己实现HttpMessageConverter用别的Json处理lib来实现）在方法上使用注解@ResponseBody 来返回json。   但是遇到一个问题，我们返回的对象，比如User，是有很多属性的，并不需要序列化所有的属性，而且是每个方法上不一样的（所以不能直接在实体上写Jackson的ignore注解来实现），想在controller层的方法上加个自定义注解来实现，但是HttpMessageConverter这个接口，并没有传入任何当前的handlerMethod，所以也就无法获取当前方法上的我自己定义的注解。   不知道有没有什么好的方案来实现这样的功能？http://www.iteye.com/images/smiles/icon_biggrin.gif  customModelAndViewResolvers倒是可以实现这个功能，但是感觉不太方便 | |
| [返回顶楼](http://www.iteye.com/topic/1122937) | | [回帖地址](http://www.iteye.com/topic/1122937#2343540)  [0](http://www.iteye.com/topic/1122937) [0](http://www.iteye.com/topic/1122937) 已投票 | |
|  | | | |
| * **key232323** * 等级: 五星会员 * [key232323的博客](http://key232323.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 838 * 积分: 780 * 来自: 深圳 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-21  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=key232323%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%232343619)  **guofengcn 写道**  **guofengcn 写道**  问一个问题：   我想扩展MappingJacksonHttpMessageConverter（或者自己实现HttpMessageConverter用别的Json处理lib来实现）在方法上使用注解@ResponseBody 来返回json。   但是遇到一个问题，我们返回的对象，比如User，是有很多属性的，并不需要序列化所有的属性，而且是每个方法上不一样的（所以不能直接在实体上写Jackson的ignore注解来实现），想在controller层的方法上加个自定义注解来实现，但是HttpMessageConverter这个接口，并没有传入任何当前的handlerMethod，所以也就无法获取当前方法上的我自己定义的注解。   不知道有没有什么好的方案来实现这样的功能？http://www.iteye.com/images/smiles/icon_biggrin.gif  customModelAndViewResolvers倒是可以实现这个功能，但是感觉不太方便  每个方法上不一样——是不是可以自己wrap一个JsonTransformerWrap，传递一个Object target，然后根据每个方法，再set一个List标示想output哪些properties，   然后用这个JsonTransformerWrap去做json output   PS：如果是动态语言，就更容易实现些，可以在运行时候动态修改/删除User的properties | |
| [返回顶楼](http://www.iteye.com/topic/1122937) | | [回帖地址](http://www.iteye.com/topic/1122937#2343619)  [0](http://www.iteye.com/topic/1122937) [0](http://www.iteye.com/topic/1122937) 已投票 | |
| **锁定老帖子**[**主题：Spring开闭原则的表现-BeanPostProcessor扩展点-2**](http://www.iteye.com/topic/1122937)  **该帖已经被评为精华帖** | | | |
| **作者** | | **正文** | |
| * **jinnianshilongnian** * 等级: 五钻会员 * [jinnianshilongnian的博客](http://jinnianshilongnian.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 1112 * 积分: 2280 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-21  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=jinnianshilongnian%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%3Fpage%3D2%232343653)  **key232323 写道**  **guofengcn 写道**  **guofengcn 写道**  问一个问题：   我想扩展MappingJacksonHttpMessageConverter（或者自己实现HttpMessageConverter用别的Json处理lib来实现）在方法上使用注解@ResponseBody 来返回json。   但是遇到一个问题，我们返回的对象，比如User，是有很多属性的，并不需要序列化所有的属性，而且是每个方法上不一样的（所以不能直接在实体上写Jackson的ignore注解来实现），想在controller层的方法上加个自定义注解来实现，但是HttpMessageConverter这个接口，并没有传入任何当前的handlerMethod，所以也就无法获取当前方法上的我自己定义的注解。   不知道有没有什么好的方案来实现这样的功能？http://www.iteye.com/images/smiles/icon_biggrin.gif  customModelAndViewResolvers倒是可以实现这个功能，但是感觉不太方便  每个方法上不一样——是不是可以自己wrap一个JsonTransformerWrap，传递一个Object target，然后根据每个方法，再set一个List标示想output哪些properties，   然后用这个JsonTransformerWrap去做json output   PS：如果是动态语言，就更容易实现些，可以在运行时候动态修改/删除User的properties  1、这位仁兄所的就可以解决，自定义Transformer 如Model -----Map （指定要排除的对象图）。这种方式需要每个功能处理方法写一次转换。这种方式最简单    2、从你说ModelAndViewResolver知道使用3.0，实现起来是比较麻烦，如你需要按照如下流程自己实现：   if (AnnotationUtils.findAnnotation(handlerMethod, ResponseBody.class) != null) {  handleResponseBody(returnValue, webRequest); //此处需要你自定义  return null;  }   而且可能此处只能不能自动获取注册的MessageConverter，比较麻烦，，，所以还不如采用第一种方式。    3、如果你使用Spring3.1，可以注册HandlerMethodReturnValueHandler到RequestMappingHandlerAdapter，如RequestResponseBodyMethodProcessor用于支持@RequestBody注解的，你可以覆盖这个：        //通过returnType就可以得到handleMethod       public void handleReturnValue(Object returnValue, MethodParameter returnType,  ModelAndViewContainer mavContainer, NativeWebRequest webRequest)  throws IOException, HttpMediaTypeNotAcceptableException {   mavContainer.setRequestHandled(true); //设置成true后表示流程结束 直接返回  if (returnValue != null) {  writeWithMessageConverters(returnValue, returnType, webRequest);//写msg出去  }  }   如果你使用Spring3.1 可以比较优雅通过注解来解决。 | |
| [返回顶楼](http://www.iteye.com/topic/1122937?page=2) | | [回帖地址](http://www.iteye.com/topic/1122937?page=2#2343653)  [0](http://www.iteye.com/topic/1122937?page=2) [0](http://www.iteye.com/topic/1122937?page=2) 已投票 | |
|  | | | |
| * **guofengcn** * 等级: 初级会员 * [guofengcn的博客](http://guofengcn.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 31 * 积分: 30 * 来自: 北京 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-21  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=guofengcn%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%3Fpage%3D2%232343665)  **jinnianshilongnian 写道**  3、如果你使用Spring3.1，可以注册HandlerMethodReturnValueHandler到RequestMappingHandlerAdapter，如RequestResponseBodyMethodProcessor用于支持@RequestBody注解的，你可以覆盖这个：        //通过returnType就可以得到handleMethod       public void handleReturnValue(Object returnValue, MethodParameter returnType,  ModelAndViewContainer mavContainer, NativeWebRequest webRequest)  throws IOException, HttpMediaTypeNotAcceptableException {   mavContainer.setRequestHandled(true); //设置成true后表示流程结束 直接返回  if (returnValue != null) {  writeWithMessageConverters(returnValue, returnType, webRequest);//写msg出去  }  }   如果你使用Spring3.1 可以比较优雅通过注解来解决。  我现在用的是3.1，这个问题是3.0刚刚发布的时候发现的，后来也没用spring MVC 3，所以也就没再研究这个问题。   最近刚好有点其他想法，正好看到你的一系列帖子，感觉很好！   我用你说的第三种方式尝试一下！   多谢！http://www.iteye.com/images/smiles/icon_biggrin.gif | |
| [返回顶楼](http://www.iteye.com/topic/1122937?page=2) | | [回帖地址](http://www.iteye.com/topic/1122937?page=2#2343665)  [0](http://www.iteye.com/topic/1122937?page=2) [0](http://www.iteye.com/topic/1122937?page=2) 已投票 | |
|  | | | |
| * **jinnianshilongnian** * 等级: 五钻会员 * [jinnianshilongnian的博客](http://jinnianshilongnian.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 1112 * 积分: 2280 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-21  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=jinnianshilongnian%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%3Fpage%3D2%232343666)  **guofengcn 写道**  **jinnianshilongnian 写道**  3、如果你使用Spring3.1，可以注册HandlerMethodReturnValueHandler到RequestMappingHandlerAdapter，如RequestResponseBodyMethodProcessor用于支持@RequestBody注解的，你可以覆盖这个：        //通过returnType就可以得到handleMethod       public void handleReturnValue(Object returnValue, MethodParameter returnType,  ModelAndViewContainer mavContainer, NativeWebRequest webRequest)  throws IOException, HttpMediaTypeNotAcceptableException {   mavContainer.setRequestHandled(true); //设置成true后表示流程结束 直接返回  if (returnValue != null) {  writeWithMessageConverters(returnValue, returnType, webRequest);//写msg出去  }  }   如果你使用Spring3.1 可以比较优雅通过注解来解决。  我现在用的是3.1，这个问题是3.0刚刚发布的时候发现的，后来也没用spring MVC 3，所以也就没再研究这个问题。   最近刚好有点其他想法，正好看到你的一系列帖子，感觉很好！   我用你说的第三种方式尝试一下！   多谢！http://www.iteye.com/images/smiles/icon_biggrin.gif  成功了告诉兄弟下，或者写篇文章分享下 哈哈，我只是刚才看了下SpringMVC执行流程觉得这样可行http://www.iteye.com/images/smiles/icon_biggrin.gif ！ | |
| [返回顶楼](http://www.iteye.com/topic/1122937?page=2) | | [回帖地址](http://www.iteye.com/topic/1122937?page=2#2343666)  [0](http://www.iteye.com/topic/1122937?page=2) [0](http://www.iteye.com/topic/1122937?page=2) 已投票 | |
|  | | | |
| * **guofengcn** * 等级: 初级会员 * [guofengcn的博客](http://guofengcn.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 31 * 积分: 30 * 来自: 北京 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-21  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=guofengcn%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%3Fpage%3D2%232343672)  **jinnianshilongnian 写道**  **guofengcn 写道**  **jinnianshilongnian 写道**  3、如果你使用Spring3.1，可以注册HandlerMethodReturnValueHandler到RequestMappingHandlerAdapter，如RequestResponseBodyMethodProcessor用于支持@RequestBody注解的，你可以覆盖这个：        //通过returnType就可以得到handleMethod       public void handleReturnValue(Object returnValue, MethodParameter returnType,  ModelAndViewContainer mavContainer, NativeWebRequest webRequest)  throws IOException, HttpMediaTypeNotAcceptableException {   mavContainer.setRequestHandled(true); //设置成true后表示流程结束 直接返回  if (returnValue != null) {  writeWithMessageConverters(returnValue, returnType, webRequest);//写msg出去  }  }   如果你使用Spring3.1 可以比较优雅通过注解来解决。  我现在用的是3.1，这个问题是3.0刚刚发布的时候发现的，后来也没用spring MVC 3，所以也就没再研究这个问题。   最近刚好有点其他想法，正好看到你的一系列帖子，感觉很好！   我用你说的第三种方式尝试一下！   多谢！http://www.iteye.com/images/smiles/icon_biggrin.gif  成功了告诉兄弟下，或者写篇文章分享下 哈哈，我只是刚才看了下SpringMVC执行流程觉得这样可行http://www.iteye.com/images/smiles/icon_biggrin.gif ！  OK，技术在于分享！http://www.iteye.com/images/smiles/icon_biggrin.gif   你发的这一系列文章都很好，继续...我们都时刻关注着呢！（虽然有时候光看不回帖http://www.iteye.com/images/smiles/icon_redface.gif ） | |
| [返回顶楼](http://www.iteye.com/topic/1122937?page=2) | | [回帖地址](http://www.iteye.com/topic/1122937?page=2#2343672)  [0](http://www.iteye.com/topic/1122937?page=2) [0](http://www.iteye.com/topic/1122937?page=2) 已投票 | |
|  | | | |
| * **jinnianshilongnian** * 等级: 五钻会员 * [jinnianshilongnian的博客](http://jinnianshilongnian.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 1112 * 积分: 2280 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-21  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=jinnianshilongnian%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%3Fpage%3D2%232343673)  **guofengcn 写道**  **jinnianshilongnian 写道**  **guofengcn 写道**  **jinnianshilongnian 写道**  3、如果你使用Spring3.1，可以注册HandlerMethodReturnValueHandler到RequestMappingHandlerAdapter，如RequestResponseBodyMethodProcessor用于支持@RequestBody注解的，你可以覆盖这个：        //通过returnType就可以得到handleMethod       public void handleReturnValue(Object returnValue, MethodParameter returnType,  ModelAndViewContainer mavContainer, NativeWebRequest webRequest)  throws IOException, HttpMediaTypeNotAcceptableException {   mavContainer.setRequestHandled(true); //设置成true后表示流程结束 直接返回  if (returnValue != null) {  writeWithMessageConverters(returnValue, returnType, webRequest);//写msg出去  }  }   如果你使用Spring3.1 可以比较优雅通过注解来解决。  我现在用的是3.1，这个问题是3.0刚刚发布的时候发现的，后来也没用spring MVC 3，所以也就没再研究这个问题。   最近刚好有点其他想法，正好看到你的一系列帖子，感觉很好！   我用你说的第三种方式尝试一下！   多谢！http://www.iteye.com/images/smiles/icon_biggrin.gif  成功了告诉兄弟下，或者写篇文章分享下 哈哈，我只是刚才看了下SpringMVC执行流程觉得这样可行http://www.iteye.com/images/smiles/icon_biggrin.gif ！  OK，技术在于分享！http://www.iteye.com/images/smiles/icon_biggrin.gif   你发的这一系列文章都很好，继续...我们都时刻关注着呢！（虽然有时候光看不回帖http://www.iteye.com/images/smiles/icon_redface.gif ）  希望有共鸣的 有想法的 咱一起探讨学习下，这样咱才会进步 呵呵。 早睡早起，注意身体！http://www.iteye.com/images/smiles/icon_biggrin.gif | |
| [返回顶楼](http://www.iteye.com/topic/1122937?page=2) | | [回帖地址](http://www.iteye.com/topic/1122937?page=2#2343673)  [0](http://www.iteye.com/topic/1122937?page=2) [0](http://www.iteye.com/topic/1122937?page=2) 已投票 | |
|  | | | |
| * **jengd** * 等级: 初级会员 * [jengd的博客](http://jengd.iteye.com/) * 文章: 6 * 积分: 60 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-04-24  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=jengd%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%3Fpage%3D2%232344842)  学习了，期待楼主更多的大作 | |
| [返回顶楼](http://www.iteye.com/topic/1122937?page=2) | | [回帖地址](http://www.iteye.com/topic/1122937?page=2#2344842)  [0](http://www.iteye.com/topic/1122937?page=2) [0](http://www.iteye.com/topic/1122937?page=2) 已投票 | |
|  | | | |
| * **lvjun106** * 等级: 初级会员 * [lvjun106的博客](http://lvjun106.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 98 * 积分: 70 * 来自: 芜湖 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-05-16  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=lvjun106%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%3Fpage%3D2%232352457)  一直关注楼主的头像，敢问楼主是涛弟弟吗？ | |
| [返回顶楼](http://www.iteye.com/topic/1122937?page=2) | | [回帖地址](http://www.iteye.com/topic/1122937?page=2#2352457)  [0](http://www.iteye.com/topic/1122937?page=2) [0](http://www.iteye.com/topic/1122937?page=2) 已投票 | |
|  | | | |
| * **jinnianshilongnian** * 等级: 五钻会员 * [jinnianshilongnian的博客](http://jinnianshilongnian.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 1112 * 积分: 2280 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-05-16  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=jinnianshilongnian%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%3Fpage%3D2%232352468)  **lvjun106 写道**  一直关注楼主的头像，敢问楼主是涛弟弟吗？  是啊，你是？ | |
| [返回顶楼](http://www.iteye.com/topic/1122937?page=2) | | [回帖地址](http://www.iteye.com/topic/1122937?page=2#2352468)  [0](http://www.iteye.com/topic/1122937?page=2) [0](http://www.iteye.com/topic/1122937?page=2) 已投票 | |
|  | | | |
| * **lvjun106** * 等级: 初级会员 * [lvjun106的博客](http://lvjun106.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 98 * 积分: 70 * 来自: 芜湖 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-05-17  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=lvjun106%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%3Fpage%3D2%232352715)  **jinnianshilongnian 写道**  **lvjun106 写道**  一直关注楼主的头像，敢问楼主是涛弟弟吗？  是啊，你是？  看ID就知道我是谁了吧！  你火了啊！ | |
| [返回顶楼](http://www.iteye.com/topic/1122937?page=2) | | [回帖地址](http://www.iteye.com/topic/1122937?page=2#2352715)  [0](http://www.iteye.com/topic/1122937?page=2) [0](http://www.iteye.com/topic/1122937?page=2) 已投票 | |
|  | | | |
| * **semmy** * 等级: 初级会员 * [semmy的博客](http://semmy.iteye.com/) * 性别: http://www.iteye.com/images/icon_minigender_1.gif * 文章: 177 * 积分: 50 * 来自: 福建 * http://www.iteye.com/images/status/offline.gif | | 发表时间：2012-05-22  [收藏](http://www.iteye.com/admin/link?user_favorite%5Btitle%5D=semmy%E5%9B%9E%E5%B8%96%3ASpring%E5%BC%80%E9%97%AD%E5%8E%9F%E5%88%99%E7%9A%84%E8%A1%A8%E7%8E%B0-BeanPostProcessor%E6%89%A9%E5%B1%95%E7%82%B9-2&user_favorite%5Burl%5D=http%3A%2F%2Fwww.iteye.com%2Ftopic%2F1122937%3Fpage%3D2%232353538)  以前有研究过，扩展点，现在没忘记了 | |
| [返回顶楼](http://www.iteye.com/topic/1122937?page=2) | | [回帖地址](http://www.iteye.com/topic/1122937?page=2#2353538)  [0](http://www.iteye.com/topic/1122937?page=2) [0](http://www.iteye.com/topic/1122937?page=2) 已投票 | |