

ringMVCframework深入总体分析 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/461/2021_2022_ringMVCfra_c104_461629.htm 在当今的MVC framework里，似乎Webwork2逐渐成为主流，Webwork2 SpringFramework的组合变得越来越流行。这似乎意味着Spring自带的MVC framework远比Webwork2差，所以大家纷纷用Webwork2来代替。确实，Spring的MVC framework不算是整个Spring的核心部件，但它的威力却超过了很多人的想象。很多人包括xiecc认为Spring的MVC framework是非常优秀的，甚至比Webwork2更优秀。下面列举一下Spring的MVC framework在设计时做出的一些重要的决定，并将之和相关的MVC framework如Webwork2或struts进行对比：一、Spring的整个MVC配置是基于IOC容器的 与struts或webwork2相比，这是一个ms有点奇怪的决定，看一下Spring MVC的配置文件，最先看到的不是action或者form，而是一些有着特定名字的bean，Bean下面的配置是一些简单或有点复杂的属性。我们看到的是机器更容易的数据结构，而不是人更容易理解的元素。但是这恰恰是Spring的MVC强大的根源！因为它的配置就是Spring的核心IOC容器的配置，这意味着所有IOC容器的威力都可以在这里展现，我们可以为所欲为地对Spring MVC进行扩展和增强，我们可以完成在其它MVC framework中很多难以想象的任务。想扩展新的URL映射方式吗？要换一个themeResolver或LocalReolver的实现吗？想在页面中显示新类型的View（比如说RDF，呵呵，一个小秘密：xiecc是研究语义网的，虽然成天不务正业，不写论文，只写八卦）？甚至想直接在Controller里定

义AOP吗？这些对Spring的MVC来说都是小菜一碟。我没有仔细研究过Webwork2的扩展机制，我知道通过Webwork2的interceptor机制，可以进行很多的扩展，甚至有一个简单简单的IOC容器。但不管它有多强大，提供了多少扩展点。它的威力都很难和真正的IOC容器相比。而struts的plugin功能则是出名的滥，虽然它也提供了plugin机制。Spring采用IOC配置的另一个原因是使Spring的MVC与Spring的IOC容器的整合变得非常的容易。Spring提供了与struts与webwork2的整合，但是这样整合都需要在进行间接的包装，感觉总不是很自然。而且还会导致一个概念多个配置，webwork2就需要在Spring里配置bean，再配置自己的xwork文件。想象一下吧，我们的bean直接就是一个controller，直接可以完成MVC的所有任务，这是多少爽的感觉。Rod Johnson采用IOC容器来实现的另一个原因是这会减少好多开发工作量。看一下urlMapping吧，它提供的property本身就是一个HashMap，只有配置完成，我们的bean里的数据就自然存在了，哈哈，好爽吧。不用象struts那样解析XML，再把它的内容一项一项地读到HashMap里。虽然这样的配置会有点怪异，但假如我们对Spring的IOC容器非常熟悉的话，会发现它非常的亲切，也非常的简单。最后是一个简单的小秘密，Spring怎么知道某个bean的配置就是urlMapping？另一个bean的配置就是viewResolver？其实很简单，把所有的bean全部读到内存里，然后通过bean的名字或类型去找就行了。通过名字去找就是简单的getBean方法，通过类型去找则使用了BeanFactoryUtils.beansOfTypeIncludingAncestors的静态方法。二、Spring提供了明确的Model，View概念和相应的数据

结构 在Spring里有一个有趣的数据类型叫做ModelAndView，它只是简单地把要显示的数据和显示的结果封装在一个类里。但是它却提供了明确的MVC概念，尤其是model概念的强化，使程序的逻辑变得更清晰了。记得以前在Struts里写程序的时候，为了显示数据经常自己把东西放到HttpSession或HttpServletRequest里（或set到form里，虽然不太有用），这造成了model概念的模糊，而且也导致了struts与JSP页面的紧耦合。假如我们要替换成Velocity，就得另外加一个plugin，因为在velocity里数据是不需要不放到request里的。Webwork2里强调的是与Web framework解耦和它的command模式的简单性，因此在它的action里只有简单的get或set方法，假如返回数据，也只是简单地返回一个String.当然这样的实现有它的好处，但是它淡化了model和view的概念。Rod Johnson认为Webwork2里的Action同时包含了Action和Model的职责，这样一个类的职责太多，不是一个很好的设计。当然Jason Carreira不太认同这种观点，因为Action里的model对象完成可以delege给其它对象。但不管怎样，这种争论的根源在于Webwork2里淡化了model，view甚至web的概念。仁者见仁，智者见智，最后的结果还是看个人喜欢好吧。

三、Spring的Controller是Singleton的，或者是线程不安全的 和Struts一样，Spring的Controller是Singleton的，这意味着每个request过来，系统都会用原有的instance去处理，这样导致了两个结果：我们不用每次创建Controller，减少了对对象创建和垃圾收集的时间；由于只有一个Controller的instance，当多个线程调用它的时候，它里面的instance变量不是线程安全的。这也是Webwork2吹嘘的地方，它的每个Action都是线程安全的。

因为每过来一个request，它就创建一个Action对象。由于现代JDK垃圾收集功能的效率已经不成问题，所以这种创建一个对象就扔掉的模式也得到了好多人的认可。Rod Johnson甚至以此为例证明J2EE提供的object pool功能是没多大价值的。但是当人们在吹嘘线程安全怎么怎么重要的时候，我想请问有多少人在多少情况下需要考虑线程安全？Rod Johnson在分析EJB的时候也提出过其它问题，并不是没有了EJB的线程安全魔法，世界就会灭亡的，大多数情况下，我们根本不需要考虑线程安全的问题，也不考虑object pool.因为我们大多数情况下不需要保持instance状态。至少我写了那么多的struts Action，写了那么多的Spring Controller，几乎没有碰到需要在instance变量保持状态的问题。当然也许是我写的代码不够多，Struts的设计者Craig R. McClanahan曾经说当时他设计struts时有两个条件不成熟：当时没有测试驱动开发的概念；当时JVM的垃圾收集性能太次。假如现在重新设计的话，他也会采用每个request生成一个新对象的设计方法，这样可以解决掉线程安全的问题了。

四、Spring不象Webwork2或tapestry那样去隐藏Servlet相关的元素如HttpServletRequest或HttpServletResponse 这又是一个重要的设计决定。

在Webwork2里我们没有HttpServletRequest或者HttpServletResponse，只有getter， setter或ActionContext里数据，这样的结果导致一个干净的Action，一个与Web完全无关的Action，一个可以在任何环境下独立运行的bean.那么Webwork2的这样一个基于Command模式的Action究竟给我们带来什么？我想主要有两点：1、 它使我们的Action可以非常容易地被测试。2、 用户可以在Action里添加业务逻辑，

并被其它类重用。然而仔细跟Spring比较一下，我们就会发现这两点功能所带来的好处其实并不象我们想象的那么显着。Spring的Controller类也可以非常轻松被测试，看一下spring-mock下面的包吧，它提供的MockHttpServletRequest，MockHttpServletResponse还有其它一些类让测试Controller变得异常轻松。再看一下Action里的业务逻辑吧，Jason Carreira曾经说我们可以尽情地在Webwork2的Action里加业务逻辑，因为Action是不依赖于Web的。但是有多少人真正往Action里加业务逻辑的？大多数人都会业务逻辑delegate给另一个Service类或Manager类。因为我们很清楚，往Action里加业务逻辑会使整个体系的分层架构变得不清晰，不管怎样，Web层就是Web层，业务层就是业务层，两者的逻辑混在一起总会带来问题的。而且往Action里加业务逻辑会使用这个Action类变得庞大，Webwork2的Action是每个request都创建实例的，尽管带来的性能影响不太大，但并不表示每次都要把业务逻辑再new出来，业务逻辑在大多数的情况下应该是单例的。不把request和response展现给用户当然还会带来功能上的损失，也许一般的场合，用用webwork2提供的接口已经足够了，但有时我们必须要知道request和response才能发挥出更大的威力。比如我以前的一个项目里有一个通过递归动态生成的树状结构的页面，在jsp页面上显示递归是痛苦或不可能的，因此我用response直接write出页面，这在spring里很easy，但在webwork里可能比较难了（偶不敢肯定，偶研究得不够深，也许高手是有办法的）。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com