



Struts1与Struts2的12点区别

作者: xp9802 <http://xp9802.iteye.com>

Struts1与Struts2的12点区别

目 录

1. 未分类

1.1 Struts1与Struts2的12点区别 3

1.1 Struts1与Struts2的12点区别

发表时间: 2011-10-28 关键字: struts, jsp, java, ssh

- 1) 在Action实现类方面的对比：Struts 1要求Action类继承一个抽象基类；Struts 1的一个具体问题是使用抽象类编程而不是接口。Struts 2 Action类可以实现一个Action接口，也可以实现其他接口，使可选和定制的服务成为可能。Struts 2提供一个ActionSupport基类去实现常用的接口。即使Action接口不是必须实现的，只有一个包含execute方法的POJO类都可以用作Struts 2的Action。
- 2) 线程模式方面的对比：Struts 1 Action是单例模式并且必须是线程安全的，因为仅有Action的一个实例来处理所有的请求。单例策略限制了Struts 1 Action能做的事，并且要在开发时特别小心。Action资源必须是线程安全的或同步的；Struts 2 Action对象为每一个请求产生一个实例，因此没有线程安全问题。
- 3) Servlet依赖方面的对比：Struts 1 Action依赖于Servlet API，因为Struts 1 Action的execute方法中有HttpServletRequest和HttpServletResponse方法。Struts 2 Action不再依赖于Servlet API，从而允许Action脱离Web容器运行，从而降低了测试Action的难度。当然，如果Action需要直接访问HttpServletRequest和HttpServletResponse参数，Struts 2 Action仍然可以访问它们。但是，大部分时候，Action都无需直接访问HttpServletRequest和HttpServletResponse，从而给开发者更多灵活的选择。
- 4) 可测性方面的对比：测试Struts 1 Action的一个主要问题是execute方法依赖于Servlet API，这使得Action的测试要依赖于Web容器。为了脱离Web容器测试Struts 1的Action，必须借助于第三方扩展：Struts TestCase，该扩展下包含了系列的Mock对象（模拟了HttpServletRequest和HttpServletResponse对象），从而可以脱离Web容器测试Struts 1的Action类。Struts 2 Action可以通过初始化、设置属性、调用方法来测试。
- 5) 封装请求参数的对比：Struts 1使用ActionForm对象封装用户的请求参数，所有的ActionForm必须继承一个基类：ActionForm。普通的JavaBean不能用作ActionForm，因此，开发者必须创建大量的ActionForm类封装用户请求参数。虽然Struts 1提供了动态ActionForm来简化ActionForm的开发，但依然需要在配置文件中定义ActionForm；Struts 2直接使用Action属性来封装用户请求属性，避免了开发者需要大量开发ActionForm类的烦琐，实际上，这些属性还可以是包含子属性的Rich对象类型。如果开发者依然怀念Struts 1 ActionForm的模式，Struts 2提供了ModelDriven模式，可以让开发者使用单独的Model对象来封装用户请求参数，但该Model对象无需继承任何Struts 2基类，是一个POJO，从而降低了代码污染。
- 6) 表达式语言方面的对比：Struts 1整合了JSTL，因此可以使用JSTL表达式语言。这种表达式语言有基本对象图遍历，但在对集合和索引属性的支持上则功能不强；Struts 2可以使用JSTL，但它整合了一种更强大和灵活的表达式语言：OGNL（Object Graph Notation Language），因此，Struts 2下的表达式语言功能更加强大。
- 7) — 绑定值到视图的对比：Struts 1使用标准JSP机制把对象绑定到视图页面；Struts 2使用“ValueStack”技术，使标签库能够访问值，而不需要把对象和视图页面绑定在一起。
- 8) 类型转换的对比：Struts 1 ActionForm 属性通常都是String类型。Struts 1使用Commons-Beanutils进行类型转换，每个类一个转换器，转换器是不可配置的；Struts 2使用OGNL进行类型转换，支持基本数据类型和常用对象之间的转换。
- 9) 数据校验的对比：Struts 1支持在ActionForm重写validate方法中手动校验，或者通过整合Commons validator框架来完成数据校验。Struts 2支持通过重写validate方法进行校验，也支持整合XWork校验框架进行

校验。

10) Action执行控制的对比：Struts 1支持每一个模块对应一个请求处理（即生命周期的概念），但是模块中的所有Action必须共享相同的生命周期。Struts 2支持通过拦截器堆栈（Interceptor Stacks）为每一个Action创建不同的生命周期。开发者可以根据需要创建相应堆栈，从而和不同的Action一起使用。

11) 捕获输入：Struts1 使用ActionForm对象捕获输入。所有的ActionForm必须继承一个基类。因为其他JavaBean不能用作ActionForm，开发者经常创建多余的类捕获输入。动态Bean（DynaBeans）可以作为创建传统ActionForm的选择，但是，开发者可能是在重新描述(创建)已经存在的JavaBean（仍然会导致有冗余的javabean）。Struts 2直接使用Action属性作为输入属性，消除了对第二个输入对象的需求。输入属性可能是有自己(子)属性的rich对象类型。Action属性能够通过 web页面上的taglibs访问。Struts2也支持ActionForm模式。rich对象类型，包括业务对象，能够用作输入/输出对象。这种 ModelDriven 特性简化了taglib对POJO输入对象的引用。