

用Stripes做JavaWeb开发 PDF转换可能丢失图片或格式，建议  
阅读原文

[https://www.100test.com/kao\\_ti2020/220/2021\\_2022\\_\\_E7\\_94\\_A8Stripes\\_c104\\_220860.htm](https://www.100test.com/kao_ti2020/220/2021_2022__E7_94_A8Stripes_c104_220860.htm) 摘要: Stripes是一个以让程序员的web开发简单而高效为准则来设计的基于动作的开源Java web框架。本文将介绍Stripes与其它如Struts之类基于动作的框架的区别和其提供的一些存在于Ruby on Rails之中的简单性。Stripes是一个以让程序员的web开发简单而高效为准则来设计的基于动作的开源Java web框架。传统的Java web开发着眼于借去耦（Decoupling）来实现其灵活性，但导致多个的配置文件，额外的对象，和其他资源的分散。这些困难造成相当多的程序员的更高的学习时间和低下的效率。其结果是有些Java程序员被一些非Java的框架所吸引去了：Ruby on Rails或者Django。一些Java web框架，如Stripes，正在开始从这些非Java框架中汲取其成功经验：简单而高效的开发。本文将介绍Stripes与其它如Struts之类基于动作的框架的区别和其提供的一些存在于Ruby on Rails之中的简单性。图1是典型的用Stripes写的应用程序中的正常事件流程和组件。图1 典型Stripes流程如你所见，其流程基本上就是一个MVC框架。Stripes和其他的基于动作的框架的一个主要的区别是没有一个外部的配置文件。我们随后将看到，Stripes用annotation和约定而非配置来提高产出和减少杂乱。编写你的第一个Stripe动作（Action）让我们现在就开始通过创建Hello World例程来了解Stripes框架和理解其运作。HelloWorldAction类将提示用户输入姓氏和名字然后在另一个View里面显示，首先我们来编写controller类。  

```
public class HelloWorldAction implements ActionBean {
```

```
@ValidateNestedProperties({ @Validate(field = "firstName",
required = true, on = {"hello"}), @Validate(field = "age", required =
true, minvalue = 13, on = {"hello"}) }) private Person person. private
ActionBeanContext context. @DefaultHandler public Resolution
index() { return new ForwardResolution("Hello.jsp"). } public
Resolution hello() { return new ForwardResolution("SayHello.jsp").
} public void setPerson(String person) {this.person = person.} public
String getPerson() { return person.} public void
setContext(ActionBeanContext c) {this.context = c. } public
ActionBeanContext getContext() {return context. }} Controller类
是一个实现了Stripes特有接口ActionBean的POJO ( Plain Old
Java Object , 译注：读破粥)。所有的Stripes动作类都要实现
这一接口以让StripesDispatcher servlet在运行服务时为其注入一个
ActionBeanContext对象。ActionBeanContext对象可以让你存取的对象如request、response、和servlet context等servlet API。
大多数时候在Stripes应用中是不用读取这些底层API对象的。
ActionBeanContext类还提供当前动作的状态并可以添加信息消息和错误消息到当前动作中。ActionBeanContext的变量
和其读写方法可以放在一个基类里面，因为所有的Stripes动作
都要实现之。Controller类的其他部分对于任何Java程序员来说
都是很面熟的。有一个Person对象和其读写方法是用来读写用户的姓名给view的。虽然这仅仅是一个简单的嵌套对象
，Stripes可以通过Java集合、泛型支持、和下标化的属性来实现
更复杂完善的数据捆绑。因为Stripes可以处理复杂数据捆绑
，你的领域对象 ( Domain Object ) 可以在其他需要它们的层
重用。例如：通过Stripes你可以很容易的收集一个领域对象的
```

信息，然后用其他的POJO框架，如Hibernate或者EJB3来对其进行持久化。Person对象变量上有一个Stripes验证annotation用来保证用户在激活hello方法的时候已经输入了姓名。如果用户没有输入这两个必需的变量，原始页会被返回，并显示一个相关的错误消息。该验证只有在hello事件被申请的时候才会被激活，因为annotation的属性中指定了(on = {"hello"})。Stripes还会使用实用默认法则，根据验证方法和变量名称产生一个错误信息。例如，如果Person类的firstName变量在提交的时候没有提供，用户将看到：Person First Name is a required field. 这条消息是通过将Person.firstName进行刻读化处理后得到的。如果有必要，这些错误消息可以被重载来提供更多的客户自定义功能。 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](http://www.100test.com)