1. 领域模型，设计模式，框架
2. M1 : jsp+javaBean ,jsp做逻辑处理
3. M2 : MVC

Model（模型）是应用程序中用于处理应用程序数据逻辑的部分。通常模型对象负责在数据库中存取数据。

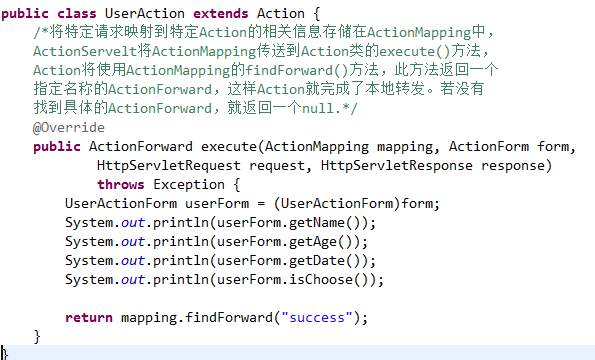
View（视图）是应用程序中处理数据显示的部分。通常视图是依据模型数据创建的。

Controller（控制器）是应用程序中处理用户交互的部分。

1. Struts1在web.xml中配置ActionServlet,所有.do结尾的请求，都交给这个Servlet，它是Struts1的控制类，url-pattern的取值，自己定义，常见的就是\*.do

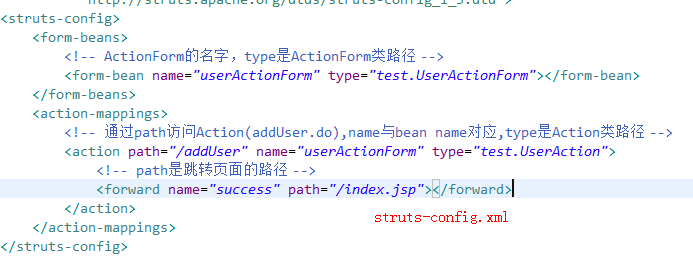


1. 创建 ActionForm
2. ActionForm 是一个JavaBean，需继承org.apache.struts.action.ActionForm 类，它捕获通过HTTP请求传送的参数
3. ActionForm 需要有无参构造方法
4. ActionForm 针对每个HTML 表单中的字段具有一个对应的属性
5. ActionServlet 匹配请求中的参数和ActionForm 中的属性，并调用ActionForm 中的setter 方法，将参数传入ActionForm
6. 创建Action
7. Action 是一个Java 类，需继承org.apache.struts.action.Action 类
8. ActionServlet 将会组装ActionForm，并将它传递给Action
9. Action 通常负责：输入校验；调用业务逻辑类执行业务逻辑操作；决定返回哪个 ActionForward



1. 配置文件

Struts用actionform封装的对象,通过name=”userActionForm”把form-bean和Action联系起来

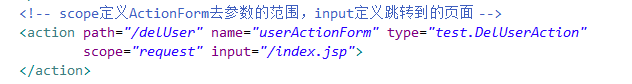


客户端重定向

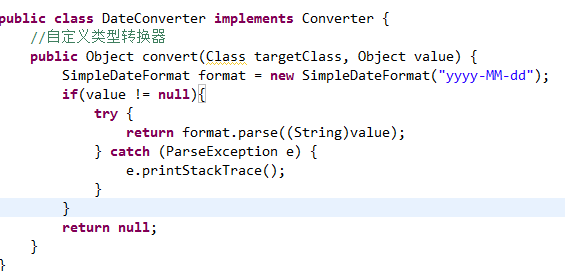


Mapping.getInputForward()跳转,定义scope是告诉Struts把ActionForm对象放到request或Session中



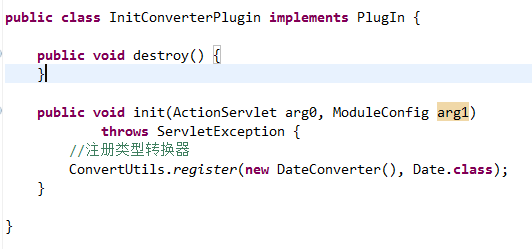


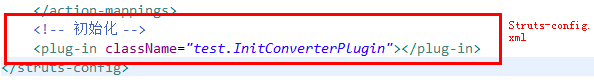
1. 时间类型处理
2. 定义类型转换器



1. 注册类型转换器

Plugin实现初始化



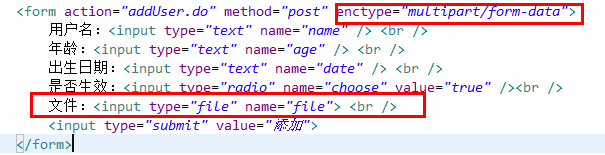


1. 文件上传
2. 拷贝依赖包

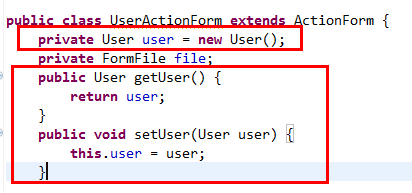




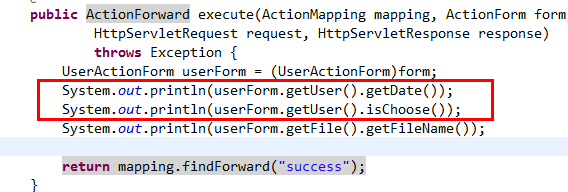
1. userForm.getFile().getFileName()



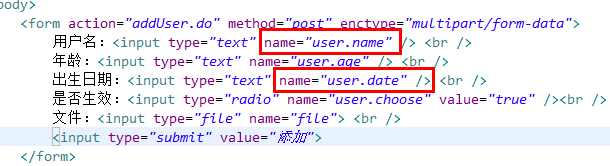
1. 向ActionForm中传递复杂对象
2. User对象作为一个属性，**user需要new**



1. 取user对象中的参数



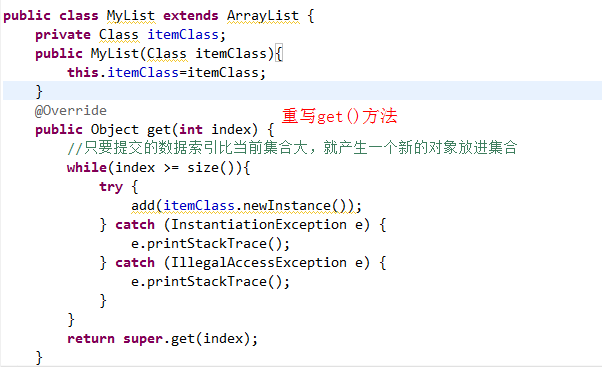
1. 表单中的user和ActionForm定义的User对象名称一致

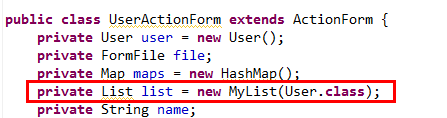


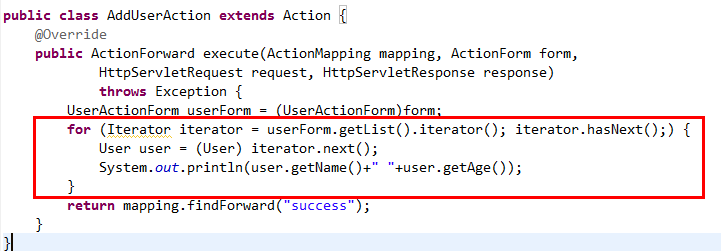
1. 向ActionForm中传递Map对象
2. private Map maps = new HashMap();
3. addr1是key，value是字符串数组，要想value是自定义类型需重写put()方法



1. 用List批量提交数据

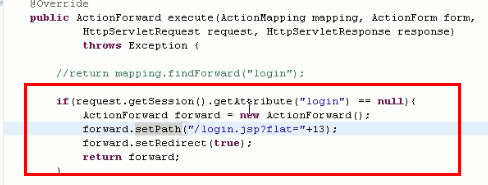








1. Session用于分步处理ActionForm数据
2. <global-forwards></global-forwards>用来定义全局的页面跳转
3. 自己定义ActionForward，定义跳转路径



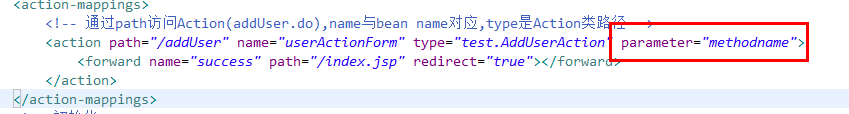
1. 继承DispatchAction,实现一个Action中写多个方法，但参数有要求
2. 拷贝jar包

struts-extras-1.3.10.jar

1. 默认调用的方法



1. 定义parameter=”参数名” 告诉Struts调用哪个方法



1. Struts1标签库
2. Bean



1. Html



1. 数据校验

DynaValidatorForm是DynaActionForm的子类，它能够提供动态ActionForm和自动表单输入验证的功能。和使用DynaActionForm类似，你必须首先在配置文件中进行配置：<form-beans>

<form-bean name="loginForm" type="org.apache.struts.validator.DynaValidatorForm">

<form-property name="actionClass" type="java.lang.String"/>

<form-property name="username" type="java.lang.String"/>

<form-property name="password" type="java.lang.String"/>

</form-bean>

</form-beans>

同时要定义验证的插件：

<plug-in className="org.apache.struts.validator.ValidatorPlugIn">

<set-property property="pathnames" value="/WEB-INF/validator-rules.xml, /WEB-INF/validation.xml"/>

</plug-in>

在Struts 1.1中也支持JavaScript客户端验证。如果你选择了客户端验证，当某个表单被提交以后，Struts 1.1启动客户端验证，如果浏览器不支持JavaScript验证，则服务器端验证被启动，这种双重验证机制能够最大限度地满足各种开发者的需要。 JavaScript验证代码也是在validator-rules.xml文件中定义的。要启动客户端验证，你必须在相应的JSP文件中做如下设置：

1. 为<html:form>增加onsubmit属性,设置Javascript支持