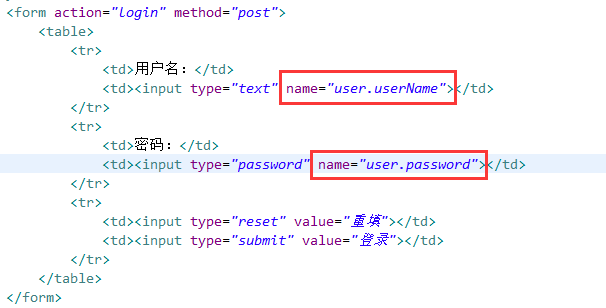
**Struts 2 接收用户输入数据的3种方式**

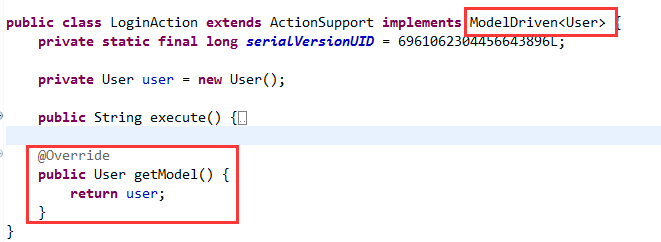
1. 使用领域对象来接收



对应的Action中有一个领域对象user，包含userName和password属性

1. Action实现ModelDriven接口



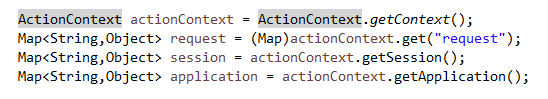


1. 使用Action的属性来接收

界面代码与第二种一致，此时的Action有JAVAbean的感觉。

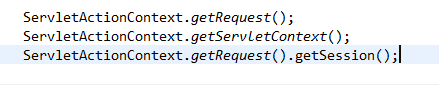
**Action中访问request、session、application对象的2种方式**

1. 松耦合方式



通过Action上下文进行

1. 耦合方式



通过ServletActionContext进行

**Struts 2框架的主要配置文件**

|  |  |  |  |
| --- | --- | --- | --- |
| 文件 | 可选 | 位置(相对于webapp) | 用途 |
| web.xml | 否 | /WEB-INF/ | 核心配置 |
| struts.xml | 是 | /WEB-INF/classes/ | result映射、action映射、拦截器配置等 |
| struts.properties | 是 | /WEB-INF/classes/ | Struts2框架属性配置 |
| struts-default.xml | 是 | /WEB-INF/lib/struts2-core.jar | Struts2提供的默认配置 |
| struts-plugin.xml | 是 | /WEB-INF/lib/struts2-xxx-plugin.jar | Struts2的插件配置 |

**Struts 2配置文件：web.xml配置**

web.xml是所有JAVA web程序的核心配置文件，需要在该文件配置Struts 2的过滤器，用于对Struts框架进行初始化，如下：

<!-- Struts2配置 -->

<filter>

<filter-name>struts</filter-name>

<filter-class>

org.apache.struts2.dispatcher.filter.StrutsPrepareAndExecuteFilter

</filter-class>

</filter>

<filter-mapping>

<filter-name>struts</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

<!-- Struts2配置 -->

**Struts 2配置文件：Struts.properties**

可参见Struts2-core-xxx.jar解压后的/org/apache/struts2/default.properties文件

**Struts2配置文件：struts.xml**

bean配置、常量(constant)配置、包(package)配置、名称空间(namespace)配置、包含(include)配置、拦截器(interceptor)配置、action配置、result配置、异常(Exception)配置

**Action配置**

Action是实现了com.opensymphony.xwork2.Action接口，默认调用execute方法；实际开发中，会继承com.opensymphony.xwork2.ActionSupport类，该类实现了Action接口和其他可选接口，提供了输入验证，错误信息存取，以及国际化支持。

**Action映射**：是将URL映射到一个action类，并执行该action的某个方法，默认是execute方法。action配置的属性有：

|  |  |  |
| --- | --- | --- |
| 属性 | 是否必须 | 说明 |
| name | 是 | action名字，匹配请求URL |
| class | 否 | action实现类的完整类名 |
| method | 否 | 执行action时调用的方法 |
| converter | 否 | 应用于action的类型转换器的完整类名 |

实例配置：

<action name=*"login"* class=*"com.wangdh.struts2.actions.LoginAction"* method=*"test"*>

<result name=*"success"*>/success.jsp</result>

<result name=*"error"*>/error.jsp</result>

</action>

**动态方法映射**:是在action的名字中使用!来标识要调用的方法，格式为：actionName!methodName.action. actionName是配置的action名字。

如当请求/news!create.action将调用NewsAction的create方法，即使在配置时指定了要执行的方法，还是将会执行新的指定的方法。

**默认action配置**：当请求一个不存在的action时，会报404错误，此时可以指定一个默认action，当没有action匹配到时，执行配置的这个默认action，如：

<package name=*"com.wangdh.struts2.actions"* extends=*"struts-default"*>

<!-- 配置默认action，当没有action配置时，执行该action -->

<default-action-ref name=*"helloWorld"*/>

<action name=*"helloWorld"* class=*"com.wangdh.struts2.actions.HelloWorldAction"*>

<result name=*"success"*>/HelloWorld.jsp</result>

</action>

<action name=*"login"* class=*"com.wangdh.struts2.actions.LoginAction"* method=*"test"*>

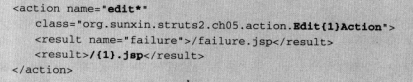
<result name=*"success"*>/success.jsp</result>

<result name=*"error"*>/error.jsp</result>

</action>

</package>

**通配符映射（没试成功）**：当web程序规模增大时，action的数量会增加，从而需要配置大量的action映射，此时可以使用通配符(\*)来进行配置.如：



请求/editNews和/editCategory会匹配到上面的配置；{N}，N从1到9，如上面的配置中{1}匹配为News/Category，{0}匹配请求的路径，即editNews.

**Result配置**

一个result代表一个可能的结果，action的方法执行完后，会返回一个字符串的结果码，框架会查找名称为该结果码的result，向用户输出。result元素的属性有：

|  |  |  |
| --- | --- | --- |
| 属性 | 是否必须 | 说明 |
| name | 否 | result的name，没有指定时为success |
| type | 否 | result的类型，代表不同类型的输出，默认是dispatcher |

Struts 2预定义的结果类型

|  |  |
| --- | --- |
| 结果类型 | 说明 |
| chain | 用于Action链式处理 |
| dispatcher | 用于web资源集成，包含JSP集成 |
| freemarker | 用于freemarker集成 |
| httpheader | 用于控制特殊的http行为 |
| redirect | 重定向到另一个URL |
| redirectAction | 重定向到另一个action |
| stream | 向浏览器返回一个inputstream，通常用于文件下载 |
| velocity | 用于velocity集成 |
| xslt | 用于xml/xslt集成 |
| plainText | 用于显示某个页面的特定内容 |

在框架的Struts-default.xml文件中给出了result-types的定义。

**全局结果**

在action中配置的结果只能在该action中使用，很多时候，重新登录，错误结果在很多action都会返回，此时，可以使用全局结果配置，避免每个action都进行配置，如下：

<!-- 全局结果配置 -->

<global-results>

<result name=*"error"*>/error.jsp</result>

<result name=*"login"* type=*"redirectAction"*>login!input</result>

</global-results>

**动态结果**

调整的页面是由程序控制的，但是返回的字符串是固定的，即result的name是固定的，配置如下：

