3、团队协作开发配置

|  |
| --- |
| <include file = *"config/qimh/struts/user.xml"*/> |

4、配置文件加载顺序

struts.xml--->struts-plugin.xml--->struts.xml

5、package的配置

|  |
| --- |
| <package name=*"user"* namespace=*"/user"* extends=*"struts-default"*>  <!-- name是请求名称，不要写/ ;class对应action完全限定名=包名+类名 -->  <action name=*"login"* class=*"com.qimh.LoginAction"*>  <!-- result 是结果集 name值和对应的action中的方法的返回值匹配，默认是success-->  <result name = *"success"*>/success.jsp</result>  <result name = *"failed"*>/login.jsp</result>  </action>  </package> |

6、action的配置

|  |
| --- |
| <!-- name是url请求名称，不要写/ ;class是处理url请求对应的java类 class要求=包名+类名  并且该类是由公共的无参构造方法，method配置处理请求类的处理方法，默认为execute ;方法满足是公共的，返回值类型是string，无参  method和name那么无关  -->    <action name=*"loginOut"* class=*"com.qimh.LoginAction"* method=*"loginOut"*>  <!-- result 是结果集 name值和对应的action中的方法的返回值匹配，默认是success-->  <result name = *"success"*>/index.jsp</result>  </action> |

7、result 配置

|  |
| --- |
| <!-- result 是结果集 name值和对应的action中的方法的返回值匹配，默认是success-->  <!-- result-指结果集  name：匹配请求处理方法返回的值，默认是success  type:结果处理类型 ，默认是dispatcher请求转发  type 有哪些：  <result-type name="chain" class="com.opensymphony.xwork2.ActionChainResult"/>  <result-type name="dispatcher" class="org.apache.struts2.dispatcher.ServletDispatcherResult" default="true"/>  <result-type name="freemarker" class="org.apache.struts2.views.freemarker.FreemarkerResult"/>  <result-type name="httpheader" class="org.apache.struts2.dispatcher.HttpHeaderResult"/>  <result-type name="redirect" class="org.apache.struts2.dispatcher.ServletRedirectResult"/>  <result-type name="redirectAction" class="org.apache.struts2.dispatcher.ServletActionRedirectResult"/>  <result-type name="stream" class="org.apache.struts2.dispatcher.StreamResult"/>  <result-type name="velocity" class="org.apache.struts2.dispatcher.VelocityResult"/>  <result-type name="xslt" class="org.apache.struts2.views.xslt.XSLTResult"/>  <result-type name="plainText" class="org.apache.struts2.dispatcher.PlainTextResult" />  <result-type name="postback" class="org.apache.struts2.dispatcher.PostbackResult" />    chain：指action链，链接下一个action，执行actionA以后直接执行actionB后执行actionC，用的较少 ，地址栏是请求的第一action  案例：<result type = "chain" name = "success">loginOut</result>  \*\*\*dispatcher：转发，和servlet一致，如果request中有数据要到视图中显示，那么使用dispatcher  案例：<result type = "dispatcher" name = "success">/success.jsp</result>  \*\*\*redirect：重定向，如果是重定向到jsp页面，可以直接重定向，如果是重定向到另外一个action，  需注意，是否配置的action的后缀名，如果要求有后缀名，那么重定向的action一定要加上后缀名  案例：<result type = "redirect" name = "success">loginOut</result>  \*\*\*redirectAction：重定向到另外一个action，不用加action的后缀，他会将前一个action的后缀名自动加上  案例：<result type = "redirectAction" name = "success">loginOut</result>  \*\*\*stream：以流的形式显示 ，文件下载    -->  <result type = *"redirectAction"* name = *"success"*>loginOut</result>  <result name = *"failed"*>/login.jsp</result> |

8、全局配置：

|  |
| --- |
| <!-- 配置全局结果集 -->  <global-results>  <result name = *"login"*>/login.jsp</result>  </global-results> |

注意：放在package里的最前面

在action的配置中，如果不去配置class属性，将会有默认的action执行，默认的action是ActionSupport类（com.opensymphony.xwork2.ActionSupport.class）

补充说明：查看配置标签(default-class-ref)-----D:\ProgramFiles\_ecplise\workspace-myecplise10\02struts2\_login\WebRoot\WEB-INF\lib\struts2-core-2.3.30.jar---->struts-2.3.dtd

|  |
| --- |
| <!-- 配置默认执行的action 当请求的action不存在时，那么执行默认的action-->  <default-action-ref name = *"default"*/> |

9、通配符配置

|  |
| --- |
| <!-- 使用通配符来配置action ，可以减少action的配置， \*表示匹配所有，占位符{1}表示第一个\*所代表的内容，{2}、{3}以此类推 -->  <action name=*"user\*"* class=*"com.qimh.AddAction"* method=*"{1}"*><!-- class="com.qimh.AddAction" method="" -->  <!-- result 是结果集 name值和对应的action中的方法的返回值匹配，默认是success-->  <result name = *"success"*>/index.jsp</result>  </action> |

10、减少action的配置还可以使用DMI（动态调用），不推荐 存在安全隐患。