**Struts2的一些有用的配置文件：**

1. **struts2-core.jar/struts-default.xml**
2. **struts2-core.jar/struts-2.5.dtd**
3. **struts2-core.jar/org.apache.struts2/default.properties**

**1.struts2\_0100\_Introduction:**

(1)在struts2自带的例子程序中复制struts.xml、web.xml，运行第一个Hello World程序

(2)<constant name="struts.devMode" value="true" />开发者模式

(3)<http://localhost:8080/struts2_0100_Introduction/hello.action> (.action可省略)

流程：当客户端浏览器发出此请求时，服务器首先读取web.xml配置文件，发现里面配置的struts2过滤器会过滤所有url，于是转到struts.xml，此url的namespace为/，则struts找到对应的包，此action为hello，则struts找到对应的action，在此action中，找到result标签中的Hello.jsp，于是转到此页面

**2.Struts2\_0200\_Namespace:**

(1)package与java的package起到同样的作用——避免重名（利用其name属性）

(2)package中namespace的作用就是在url中对应的那个路径

(3)result 默认name="success"

**3.Struts2\_0300\_Action:**

(1)自定义action的三种方法

(2)自己用的方法：extends ActionSupport

**4.Struts2\_0400\_Path:**

(1)struts2中的路径问题是根据action的路径而不是jsp路径来确定，所以尽量不要使用相对路径。

(2)使用相对路径的两种方法：base标签或myeclipse中的basePath

**5.Struts2\_0500\_ActionMethod:**

(1)可以在配置文件中配置Action的时候用method=来指定执行哪个方法

(2)也可以在url地址中动态指定:actionName!methodName

(3)前者会产生太多的action(配置文件中)，所以不推荐使用

(4)<constant name="struts.enable.DynamicMethodInvocation" value="true" />开启动态方法调用

**6.Struts2\_0600\_ActionWildcard:**

(1)使用通配符 \* 、{1} 可以简化配置

(2)约定优于配置

**7.Struts2\_0700\_ActionAttrParamInput：**

(1)给action传递参数：在对应的action中设置成员变量，并生成getter、setter，struts会自动赋值

**8.Struts2\_0800\_DomainModelParamInput：**

(1)使用Domain Model传递参数

(2)也可以使用VO（DO、DTO）先把参数传到action中，再根据这个对象构造出对应的领域对象

**9.Struts2\_0900\_ModelDrivenParamInput：**

在Action中实现ModelDriven<T>接口之后，在传递领域对象的参数时可以直接传递，struts会自动生成领域对象

**11.Struts2\_1100\_DataValidation&Error：**

(1)在action中使用addFieldError方法可以将错误信息添加到值栈中

(2)struts标签库的使用

(3)<s:fieldError>标签的使用（带格式）

(4)<s:debug> 结合 <s:property> 获取值栈中的值，可以拿到不带格式的错误信息

**12.Struts2\_1200\_AccessWebElements:**

(1)在action中获得Map类型的request、session、application（LoginAction1）

(2)推荐使用的获取WebElements的方式：

实现RequestAware、SessionAware、ApplicationAware接口，并实现set方法。

这种方法的思想是IoC（控制反转），本应该由程序员完成的操作直接交给了struts去完成。

(3)获取HttpRequesti、HttpSession、ServletContext的两种方法

**13. Struts2\_1300\_IncludeModules:**

(1)可以使用<include file=”login.xml”/>标签将struts配置文件分解成多个模块

**14. Struts2\_1400\_DefaultAction:**

(1)<default-action-ref name=*"index"*></default-action-ref>

用这个标签来设置默认的action

**15. Struts2\_1500\_ResultType:**

(1)<result type=”dispatcher | redirect | chain | redirectAction”

**16. Struts2\_1600\_GlobalResult:**

(1) <global-results>标签的使用

(2)可以使用package中的extends属性继承其他package的配置

**17. Struts2\_1700\_DynamicResult:**

(1) <result>${r}</result> 可以使用这种方式来引用action中的成员变量

**18. Struts2\_1800\_ResultWithParams**:

(1)当时用redirect的方式时，可以向result中传递action的参数：

<result type=*"redirect"*>/user\_success.jsp?t=${type}</result>

**19.Struts2\_1900\_OGNL:**

(1)ognl的各种用法

**20. Struts2\_2000\_StrutsTags:**

(1)struts中的标签库

**21. Struts2\_2100\_Theme\_1:**

(1) 在struts.xml中控制theme，默认为xhtml，可以设置为：simple/css\_html/ajax

**22. Struts2\_2100\_Theme\_2:**

(1)用css去掉struts2中fieldError标签的格式

**23. Struts2\_2100\_Theme\_3:**

(1)通过覆盖struts2自带的格式文件来去掉fieldError的格式

**24. Struts2\_2100\_Theme\_4:**

(1)定义自己的template

**bbs后续**

**1. struts2\_3300\_interceptor:**

(1)这个项目是用来读Struts2的filter源码的

**2. struts2\_3400\_interceptor\_simulation:**

(1)这个项目是用来模拟struts2中拦截器的调用过程的

**3. struts2\_3500\_my\_interceptor:**

(1)自定义拦截器

**4. struts2\_3600\_token\_interceptor:**

(1)防止表单重复提交的token拦截器

**5. struts2\_3700\_type\_conversion:**

(1)介绍struts中自动完成的类型转换

(2)自定义类型转换器