

Evan

Only let oneself become strong enough, good enough, can afford the life that you want to.

目录视图

摘要视图

订阅

从创业到再就业，浅谈对程序员职业生​​涯的看法 征文 | 你会为 AI 转型么？ 赠书：7月大咖新书机器学习/Android/python

struts2: config-browser-plugin 与 convention-plugin 学习

2015-12-01 16:37 290人阅读 评论(0) 收藏 举报

分类： web后台 (12)

struts2被很多新手诟病的一个地方在于“配置过于复杂”，相信不少初学者因为这个直接改投spring-MVC了。convention-plugin、config-browser-plugin这二个插件的出现，很大程度改善了这个困境。

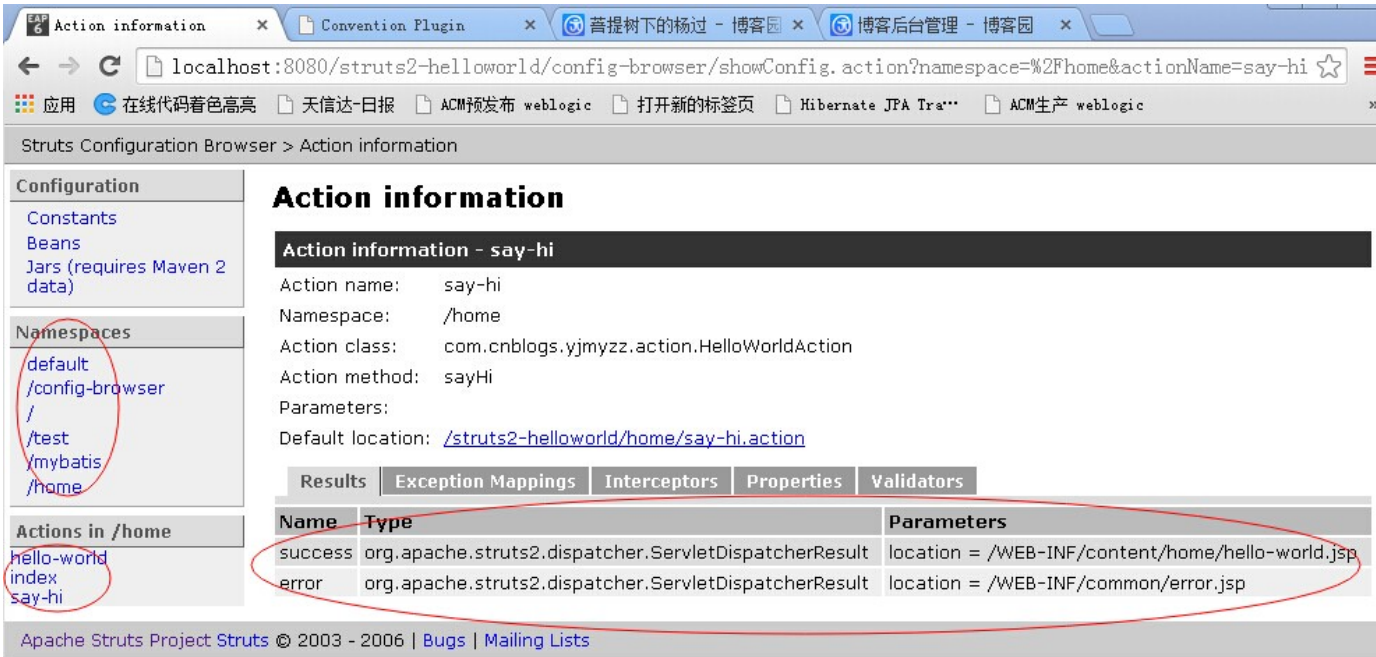
简言之：convention-plugin采用“约定大于配置”的思想，只要我们遵守约定，完全可以少写配置甚至不写配置；而config-browser-plugin则用于方便的浏览项目中的所有action及其与jsp view的映射。这二个插件结合起来学习，能很方便的搞定struts2中各种复杂的action-view映射需求。

一、config-browser-plugin使用

```
[html]
01. <span style="font-size:14px;">
02.   <dependency>
03.     <groupId>org.apache.struts</groupId>
04.     <artifactId>struts2-config-browser-plugin</artifactId>
05.     <version>2.3.16</version>
06.   </dependency>
07.
08. </span>
```

maven项目的pom.xml中加上这个即可，运行后，浏览 http://localhost:8080/{你的项目名称}/config-browser/ 即可看到当前项目中的所有action

注：以下内容中，凡有url的地方，项目名称假设为struts2-helloworld



如果跑不起来，检查服务器应用WEB-INF/lib/下是否有struts2-config-browser-plugin-2.3.16.jar 这个文件

二、convention-plugin 使用

[html]

```
01. <span style="font-size:14px;"><dependency>
02.   <groupId>org.apache.struts</groupId>
03.   <artifactId>struts2-convention-plugin</artifactId>
04.   <version>2.3.16</version>
05. </dependency></span>
```

pom.xml中加上这个后，可以把struts.xml配置文件给干掉了（或者改个名），部署App，如果启动正常，则表示环境ok。如果提示缺少asm 啥类，检查下面这几个依赖项，是否也加进来了

[html]

```
01. <span style="font-size:14px;"><dependency>
02.   <groupId>asm</groupId>
03.   <artifactId>asm</artifactId>
04.   <version>3.3.1</version>
05. </dependency>
06.
07. <dependency>
08.   <groupId>asm</groupId>
09.   <artifactId>asm</artifactId>
10.   <version>3.3.1</version>
11. </dependency>
12.
13. <dependency>
14.   <groupId>asm</groupId>
15.   <artifactId>asm-commons</artifactId>
16.   <version>3.3.1</version>
17. </dependency>
18.
19. <dependency>
20.   <groupId>asm</groupId>
21.   <artifactId>asm-tree</artifactId>
22.   <version>3.3.1</version>
23. </dependency></span>
```

2.1 零action的view

convention-plugin约定所有的jsp view都放在WEB-INF/content目录下，在这个目录下先随便放一个名为"no-action.jsp"的jsp文件，里面随便写点啥

浏览 <http://localhost:8080/struts2-helloworld/no-action>



no-action.jsp

即：即使没有对应的Action类，struts2也能按约定正常展现页面。（当然，这只是开胃小菜，真正应用中，除了做一些纯静态的页面原型之外，大部分场景，背后还是要有Action类来支撑的）

2.2 常规映射

建一个HelloWorld.action类

[java]

```
01. <span style="font-size:14px;">package com.cnblogs.yjmyzz.action;
02.
03. import org.apache.struts2.convention.annotation.Action;
04. import org.apache.struts2.convention.annotation.Namespace;
05. import org.apache.struts2.convention.annotation.Result;
06.
07. @Namespace("/home")
08. public class HelloWorldAction extends BaseAction {
```

```

09.
10.     private static final long serialVersionUID = -8827776224243873974L;
11.
12.     private String message;
13.
14.     @Action("hello-world")
15.     public String execute() throws Exception {
16.         return SUCCESS;
17.     }
18.
19.     @Action(value = "say-hi", results = { @Result(name = "success", location = "hello-world.jsp") })
20.     public String sayHi() throws Exception {
21.         message = "welcome to SSH!";
22.         return SUCCESS;
23.     }
24.
25.     public String getMessage() {
26.         return message;
27.     }
28.
29.     public void setMessage(String message) {
30.         this.message = message;
31.     }
32.
33. }</span>

```

解释一下：

第7行，在整个Action类上使用了`@Namespace("/home")`，表示整个Action最终浏览的url，是以 `http://localhost:8080/{你的项目名称}/home/` 打头

第14行，通过注解`@Action("hello-world")`，把默认的`/home/index.action`路径，改成了 `/home/hello-world`

至于execute方法，返回success后，对应的是哪个jsp文件，这个不用死记，通过config-browser-plugin看下便知

Configuration

- Constants
- Beans
- Jars (requires Maven 2 data)

Namespaces

- default
- /config-browser
- /home
- /test
- /mybatis

Actions in /home

- hello-world
- say-hi

Action information

Action information - hello-world

Action name: hello-world
 Namespace: /home
 Action class: com.cnblogs.yjmyzz.action.HelloWorldAction
 Action method: execute
 Parameters:
 Default location: [/struts2-helloworld/home/hello-world.action](#)

Results
Exception Mappings
Interceptors
Properties
Validators

Name	Type	Parameters
input	org.apache.struts2.dispatcher.ServletDispatcherResult	location = /WEB-INF/content/home/hello-world.jsp
error	org.apache.struts2.dispatcher.ServletDispatcherResult	location = /WEB-INF/content/home/hello-world.jsp
success	org.apache.struts2.dispatcher.ServletDispatcherResult	location = /WEB-INF/content/home/hello-world.jsp

即：execute方法返回input/error/success中的任何一个，都会映射到`/WEB-INF/content/home/hello-world.jsp` 这个文件上

20行sayHi()方法上的注解有点意思，`@Action(value = "say-hi", results = { @Result(name = "success", location = "hello-world.jsp") })`，默认情况下，如果不指定location，返回success时，它应该对应 `/WEB-INF/content/home/say-hi.jsp` 这个文件，但是通过location值，变成了 `hello-world.jsp`，即与`/home/hello-world`共用同一个view。

3. 拦截器问题

上一篇学习了通过拦截器来处理异常，采用convention插件后，会发现拦截器不起作用(struts.xml中配置了也一样)

```

[html]
01. <span style="font-size:14px;"><?xml version="1.0" encoding="UTF-8" ?>
02. <!DOCTYPE struts PUBLIC
03.     "-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
04.     "http://struts.apache.org/dtds/struts-2.3.dtd">
05.
06. <struts>
07.
08.     <constant name="struts.enable.DynamicMethodInvocation" value="false" />
09.     <constant name="struts.devMode" value="false" />

```

```

10.
11. <package name="default" namespace="/" extends="struts-default">
12.
13.     <interceptors>
14.         <interceptor name="myinterceptor"
15.             <class="com.cnblogs.yjmyzz.Interceptor.ExceptionInterceptor">
16.         </interceptor>
17.
18.         <interceptor-stack name="myStack">
19.             <interceptor-ref name="myinterceptor" />
20.         </interceptor-stack>
21.     </interceptors>
22.
23.     <default-interceptor-ref name="myStack" />
24.     <default-action-ref name="index" />
25.
26.     <global-results>
27.         <result name="error">/WEB-INF/common/error.jsp</result>
28.     </global-results>
29.
30.     <global-exception-mappings>
31.         <exception-mapping exception="java.lang.Exception"
32.             result="error" />
33.     </global-exception-mappings>
34.
35.     <action name="index">
36.         <result type="redirectAction">
37.             <param name="actionName">hello-world</param>
38.             <param name="namespace">/home</param>
39.         </result>
40.     </action>
41.
42. </package>
43.
44. <!-- 因为有Convention-plugin, 就不再需要手动写action-view的映射规则了 -->
45. <!--     <include file="struts-home.xml" />
46.     <include file="struts-mybatis.xml" /> -->
47.
48. </struts></span>

```

原因在于convention-plugin使用后，所有Action不再继承自默认defaultStack对应的package，为了解决这个问题，建议所有Action都继承自一个自定义的BaseAction类，然后在BaseAction上加上@ParentPackage("default")，让所有Action强制继承自struts.xml中定义的default package

[java]

```

01. <span style="font-size:14px;">package com.cnblogs.yjmyzz.action;
02.
03. import org.apache.struts2.convention.annotation.ParentPackage;
04. import org.slf4j.Logger;
05. import org.slf4j.LoggerFactory;
06.
07. import com.opensymphony.xwork2.ActionSupport;
08.
09. @ParentPackage("default")
10. public class BaseAction extends ActionSupport {
11.
12.     private static final long serialVersionUID = -4320398837758540242L;
13.     protected Logger logger = LoggerFactory.getLogger(this.getClass());
14.
15. }</span>

```

基本用法就是这些，如果不清楚Action最终出来的url是啥，或者不清楚某个url最终会对应到哪个jsp文件，无需死记规则，只要善用config-browser-plugin，大多数下不用查阅文档即可快速解决问题。

作者：菩提树下的杨过

出处：<http://yjmyzz.cnblogs.com>

本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

另外说一下我遇到的问题：缺少asm三个架包所导致（asm-3.3.jar，asm-commons-3.3.jar，asm-tree-3.3.jar）一直404错误，还有就是我犯了一个严重的错误：没看错误提示，导致我在哪里啥弄了一天，痛苦。。。