

Spring 整合 Struts2 详解

1. 整合目标：使 Spring 的 IOC 容器管理 Struts2 的 Action

2. 步骤概览：

1) . 在 WEB 应用中加入 Struts2

- ①. 加入 jar 包
- ②. 在 web.xml 文件中进行一些配置
- ③. 加入 Struts2 自身的配置文件

2) . 在 WEB 应用中加入 Spring

- ①. 加入 jar 包
- ②. 加入 Spring 的配置文件
- ③. 在 web.xml 文件中配置启动 IOC 容器的 Listener

3) . 整合

- ①. 加入 Spring 插件
- ②. 在 Spring 的配置文件中配置 Struts2 的 Action 实例
- ③. 在 Struts 配置文件中配置 action

4) . 原理分析

3. 在 WEB 应用中加入 Struts2

1) . 参照 struts-2.3.15.3-all\struts-2.3.15.3\apps 下的 struts2-blank.war

2) . 解压 struts-2.3.15.3-all\struts-2.3.15.3\apps 下的 struts2-blank.war 到当前文件夹

3) . 加入 jar 包：把 struts2-blank\WEB-INF\lib*.jar 复制到当前 WEB 应用的 WEB-INF 的 lib 目录下

4) . 打开 struts-2.3.15.3\apps\struts2-blank\WEB-INF\web.xml 文件，复制 filter 部分到当前 WEB 应用的 web.xml 文件中

```
<filter>
    <filter-name>struts2</filter-name>
    <filter-class>
        org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter
    </filter-class>
</filter>

<filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

5) . 复制 struts-2.3.15.3\apps\struts2-blank\WEB-INF\classes

下的 struts.xml 文件到当前 WEB 应用的 src 目录下，并删除部分配置，只保留如下内容：

```
<struts>

    <constant name="struts.enable.DynamicMethodInvocation" value="false" />
    <constant name="struts.devMode" value="true" />

    <package name="default" namespace="/" extends="struts-default">

        </package>

    </struts>
```

4. 在 WEB 应用中加入 Spring:

1) . 加入 Spring 的 jar 包：注意：需要加入 WEB 应用的 spring-web-4.0.0.RELEASE.jar，因为 Struts2 中已经包含了 commons-logging 的 jar 包，所以这里可以不添加 commons-logging

```
spring-beans-4.0.0.RELEASE.jar
spring-context-4.0.0.RELEASE.jar
spring-core-4.0.0.RELEASE.jar
spring-expression-4.0.0.RELEASE.jar
```

```
spring-web-4.0.0.RELEASE.jar
```

2) . 在类路径下加入 Spring 的配置文件（也可以放在其它路径下）

3) . 在 web.xml 文件中配置启动 IOC 容器的 Listener

i. 配置 Spring 配置文件的位置

```
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath:applicationContext.xml</param-value>
</context-param>
```

ii. 配置启动 IOC 容器的 Listener

```
<listener>
    <listener-class>
        org.springframework.web.context.ContextLoaderListener
```

```
</listener-class>
</listener>
```

5. Spring 整合 Struts2:

1) . 加入 Spring 插件：复制 struts-2.3.15.3-all\struts-2.3.15.3\lib\struts2-spring-plugin-2.3.15.3.jar 到当前 WEB 应用的 lib 下

2) . 在 Spring 的配置文件中配置 Struts2 的 Action 实例，注意：需要配置 scope 属性为 prototype，因为 Struts2 的 Action 不是单例的！

例如：

```
<bean id="userAction"
    class="com.atguigu.ss.action.UserAction" scope="prototype">
    <property name="userService" ref="userService"></property>
</bean>
```

3) . 在 Struts 配置文件中配置 action，但其 class 属性不再指向该 Action 的实现类，而是指向 Spring 容器中 Action 实例的 ID

例如：

```
<action name="helloworld" class="userAction">
    <result>/success.jsp</result>
</action>
```

6. Spring 整合 Struts2 原理解析：

Struts2 与 Spring 的集成要用到 Spring 插件包 struts2-spring-plugin-x-x-x.jar。Spring 插件是通过覆盖 Struts2 的 ObjectFactory 来增强核心框架对象的创建。当创建一个对象的时候，它会用 Struts2 配置文件中的 class 属性去和 Spring 配置文件中的 id 属性进行关联，如果能找到，则由 Spring 创建，否则由 Struts 2 框架自身创建，然后由 Spring 来装配。

struts2-spring-plugin-2.3.15.3.jar 下的 struts-plugin.xml 文件的代码片段：

```
<bean type="com.opensymphony.xwork2.ObjectFactory" name="spring"
class="org.apache.struts2.spring.StrutsSpringObjectFactory" />
```

具体执行过程：

```
StrutsSpringObjectFactory(SpringObjectFactory).buildBean(String, Map<String,Object>, boolean) line: 151
StrutsSpringObjectFactory(ObjectFactory).buildBean(String, Map<String,Object>) line: 151
StrutsSpringObjectFactory(ObjectFactory).buildAction(String, String, ActionConfig, Map<String,Object>) line: 121
DefaultActionInvocation.createAction(Map<String,Object>) line: 297
```