Java技巧

表单重复提交

Serverlet

拦截器

目录

[Spring初始化问题 1](#_Toc235594439)

[Spring（1）Ioc容器启动初始化上下文 2](#_Toc235594440)

[一个struts拦截器 4](#_Toc235594441)

[spring/struts2 整合 5](#_Toc235594442)

[关于struts2无法加载struts.properties的问题 7](#_Toc235594443)

[关于Illegal attempt to associate a collection with two open sessions 8](#_Toc235594444)

[struts2的datetimepicker标签的时候遇到了无法正常显示的问题 8](#_Toc235594445)

[没有解决的问题 8](#_Toc235594446)

# Spring初始化问题

1,少了antlr-3.1.jar包，下载以后会出现这样的问题

java.lang.NoSuchFieldError: ruleMemo

at org.springframework.expression.spel.generated.SpringExpressionsParser.<init>(SpringExpressionsParser.java:102)

at org.springframework.expression.spel.antlr.SpringExpressionsParserExtender.<init>(SpringExpressionsParserExtender.java:37)

http://www.antlr.org/download/antlr-3.1.jar

到这里下载   
http://www.springsource.com/repository/app/bundle/version/detail?name=com.springsource.org.antlr&version=3.0.1   
就OK了

解决方案

移出下载的antlr-3.1.jar包，下载com.springsource.org.antlr包。

然后spring就可以初始化了

# Spring（1）Ioc容器启动初始化上下文

2008-04-17 17:18

|  |
| --- |
| 1、Tomcat启动时候需要从web-xml文件读取启动配置参数，而org.springframework.web.context.ContextLoaderServlet作为Spring启动的入口，建立IOC容器体系。  （1）首先建立一个ClassLoader对象--不仅对web.xml中配置Spring的信息内容封装，而且还是根上下文（WebApplicationContext）初始化的操作类。其实ContextLoaderServlet也可以完成根上下文初始化的工作，但由于Spring提供两种WebApplicationContext加载的方式，所有设计上必须有一个通用的操作类作为Servlet的组合。Spring可以通过（Servlet）org.springframework.web.context.ContextLoaderServlet和Listener（org.springframework.web.context.ContextLoaderListener）两个类作为Spring启动的入口。  方法（一）       <context-param>           <param-name>contextConfigLocation</param-name>           <param-value>                /WEB-INF/dataAccessContext-local.xml,                 /WEB-INF/applicationContext.xml           </param-value>      </context-param>     <listener>       <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>    </listener>  方法（二）       <context-param>           <param-name>contextConfigLocation</param-name>           <param-value>                /WEB-INF/dataAccessContext-local.xml,                 /WEB-INF/applicationContext.xml           </param-value>      </context-param>       <servlet>           <servlet-name>SpringContextServlet</servlet-name>             <servlet-class>org.springframework.web.context.ContextLoaderServlet</servlet-class>           <load-on-startup>1</load-on-startup>      </servlet>       无论ContextLoaderServlet或ContextLoaderListener都是通过ContextLoader来初始化项目的WebApplicationContext。  2、根上下文（ServletContext ）和运用上下文        Servlet通过ServletContext servletContext = getServletContext()函数获得根上下文对象（servletContext ），servletContext 可以读取web.xml配置文件的参数。  获取配置参数代码如下：        String locatorFactorySelector = servletContext.getInitParameter("locatorFactorySelector");      String parentContextKey = servletContext.getInitParameter("parentContextKey");  web-xml的配置文件如下：       <context-param>            <param-name>locatorFactorySelector</param-name>            <param-value>classpath\*:beanRefContext.xml</param-value>       </context-param>       <context-param>            <param-name>parentContextKey</param-name>            <param-value>factoryBeanId</param-value>       </context-param>       而且在Bean 定义文件中（/WEB-INF/dataAccessContext-local.xml）中必须增加name=factoryBeanId的bean定义。       <bean id="factoryBeanId" class="org.springframework.context.support.ClassPathXmlApplicationContext">       </bean>         在项目的webapp中如果引用其他webapp做为其辅助插件，如果两个项目中都用到了Spring，则用以上的配置可以加载两个项目的bean，而且可以互相引用。这个一般与ejb结合使用，以ejb项目作为系统的启动项目，而其中几个module是由Spring容器来实现，通过这个方式这几个module可以跨容器访问。  3、解析bean配置.Xml文件  （1）XmlBeanDefinitionReader通过调用loadBeanDefinitions(Resource resource)方法，将每个XML文件封装成一个document对象，而XML文件中每一行则封装成一个Element对象。  （2）XmlBeanDefinitionReader通过调用registerBeanDefinitions(doc, resource)方法，将配置文件.XML的每个bean定义内容封装成一个BeanDefinition对象。  （3）XmlBeanDefinitionReader间接通过DefaultListableBeanFactory的registerBeanDefinition(String beanName, BeanDefinition beanDefinition)方法，验证XML文件配置是否正确，如果正确，则将beanDefinition对象添加到DefaultListableBeanFactory的private final Map beanDefinitionMap = new HashMap();的元素中。  XML配置主要验证以下几点：  ●如果是bean设置延迟加载（lazyInit）为true，则该bean必须是singleton。  ●如果bean配置中定义了替代方法replaced-method的属性，则不允许定义factory-method静态工厂方法的属性。  ●如果配置中定义了替代方法replaced-method的属性，必须 保证bean类或他的父类有replaced-method定义的方法。  ●必须保证一个beanName只有一个bean的定义，如果两个同样beanName的定义，会报错。  http://hiphotos.baidu.com/aimy%5Fxmu/pic/item/835be1a94384bdee1e17a2b0.jpg |

# 一个struts拦截器

package com.ada.myinterceptor;

import java.util.Map;

import com.opensymphony.xwork2.ActionContext;

import com.opensymphony.xwork2.ActionInvocation;

import com.opensymphony.xwork2.interceptor.AbstractInterceptor;

public class AuthorInterceptor extends AbstractInterceptor {

@Override

public String intercept(ActionInvocation arg0) throws Exception {

// TODO Auto-generated method stub

ActionContext ctx=arg0.getInvocationContext();

Map session=ctx.getSession();

String user=(String)session.get("user");

if(user!=null){

return arg0.invoke();

}

return "login";

}

}

拦截器配置

<interceptors>

<interceptor name=*"author"*

class=*"com.ada.myinterceptor.AuthorInterceptor"*></interceptor>

<interceptor name=*"adada"* class=*"com.ada.myinterceptor.AdaInterceptor"*></interceptor>

</interceptors>

# spring/struts2 整合

在Struts2中整合Spring的IoC支持是一件十分简单的事情。让我们一步一步来实现：  
1）复制struts2-spring-plugin-x-x-x.jar和相应的spring.jar到/WEB-INF/lib目录下。  
2）在struts.properties中设置struts.objectFactory属性值

**struts.properties**

struts.objectFactory = spring

或者是在XML文件中进行常量配置

**struts.xml**

<struts>

       <constant name="struts.objectFactory" value="spring" />

</struts>

3）配置Spring监听器

**web.xml**

<listener>

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

</listener>

通过Spring配置来注册对象

**applicationContext.xml**

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE beans PUBLIC

       "-//SPRING//DTD BEAN//EN"

       "http://www.springframework.org/dtd/spring-beans.dtd">

<beans default-autowire="byName">

       <bean id="personManager" class="com.acme.PersonManager"/>

</beans>

当然你也可以拥有更多的Spring配置文件。在web.xml中进行下列设置，从而使Spring的ApplicationContext通过匹配所给定模式的文件来初始化对象

<!-- 用来定位Spring XML文件的上下文配置 -->

<context-param>

       <param-name>contextConfigLocation</param-name>

       <param-value>/WEB-INF/applicationContext-\*.xml,classpath\*:applicationContext-\*.xml</param-value>

</context-param>

4）修改你的Struts配置文件

**struts.xml**

<!DOCTYPE struts PUBLIC

       "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"

       "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>

       <include file="struts-default.xml"/>

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

           <action name="foo" class="com.acme.Foo">

               <result>foo.ftl</result>

           </action>

       </package>

       <package name="secure" namespace="/secure" extends="default">

           <action name="bar" class="bar">

               <result>bar.ftl</result>

           </action>

       </package>

</struts>

默认情况下，Spring从下面的文件中寻找为bar所做的配置

**applicationConext.xml**

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE beans PUBLIC

       "-//SPRING//DTD BEAN//EN"

       "http://www.springframework.org/dtd/spring-beans.dtd">

<beans default-autowire="byName">

       <bean id="bar" class="com.my.BarClass" singleton="false"/>

       ...

</beans>

5）好了，现在你的Struts2和Spring就能正常的一起工作了。有几个配置技术点需要详细说明下：装配模式。你可以通过设置修改struts.properties中下列属性的值来改变装配模式。

struts.objectFactory.spring.autoWire = type

该参数可供选择的设置值有

|  |  |
| --- | --- |
| **name** | 按照你的action的属性的名字和Spring里的bean的名字匹配，如果匹配就自动装配。这是**缺省的** |
| **type** | 按照你的action的属性的类型，在Spring注册的bean中查找，如果相同就自动装配。这需要你在Spring中仅注册了一个此类型的bean |
| **auto** | Spring会试图自动监测来找到最好的方法自动装配你的action |
| **constructor** | Spring会自动装配bean的构造函数的参数 |

是否使用类缓存。你可以通过设置修改struts.properties中下列属性的值来改变是否使用Spring自身的类缓存机制。可以设定的值为true或false，默认为true。

|  |
| --- |
| 关于struts2无法加载struts.properties的问题 |
| http://blog.chinaunix.net/templates/default/images/right_line.gif |
| |  | | --- | | 最近整合了struts2和spring的问题，其中看到一个struts2的文档中写得使用struts2的ioc时要在项目下加载struts.properties.它的内容是:          struts.objectFactory = spring  但是我加入了该文件之后发现了下面得错误  严重: Exception starting filter struts2 Cannot **locate****the****chosen****ObjectFactory****implementation: spring**- [unknown location]  出现这个错误的原因是项目中少了struts2-**spring**-plugin-2.0.6.jar包，加上该包就没问题了。 | |

# 关于Illegal attempt to associate a collection with two open sessions

解决方案

主要是由于一次操作没有关闭会话，所以会出现异常，养成好的习惯，每次把相应的操作实现以后，释放掉需要关闭的东西。

网络解决思想

其实问题在于你的两个操作是否是在一个事务里，通过查看spring的事务那部分代码可以得知，如果方法在事务中，那么方法中所有的session都是同一个，这个session和对应的transaction会组装成一个sessionHolder类然后放到当前线程中，如果不在一个事务中，那么每次调用dao都会得到一个新的session，这样两次得到同一个object会使这个object和两个session关联，具体的原因就是这样的，楼主不防对照着看一下，我觉得象楼主这个操作其实应该是放到同一个事务中的，在楼主的contraller不被包含在事务中，所以才会出现这个异常的吧，另，这个异常和1:N没有什么关系的，你可以查看一下spring的sessionfactoryutil类和hibernatetransactionmanager类就明白了，我可能讲的也不是很清楚

# struts2的datetimepicker标签的时候遇到了无法正常显示的问题

一些朋友在使用struts2的datetimepicker标签的时候遇到了无法正常显示的问题,我这里谈谈我的小体会.希望能帮助到大家.代码中<s:head theme="ajax"/> 很关键,如果没有,则无法正常显示,还有在type="time"时,显示小时的数字会消失,那是因为中文的问题.在标签中加入language="en-us" 即可,如果大家有兴趣,欢迎留言讨论,谢谢.

<%@ page language="java" pageEncoding="UTF-8"%> <%@ taglib prefix="s" uri="/struts-tags" %> <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"> <html> <!-- 使用struts 2的 datetimepicher 必须有这个头文件 --> <s:head theme="ajax"/> <s:form> <s:datetimepicker name="order.data" label="购买日期" toggleType="fade" value="today"> </s:datetimepicker> <br> 时间选择组件 <s:datetimepicker name="start" label="选择出发时间" toggleType="fade" type="time" value="12:00" language="en-us"/> </s:form>

### [struts2.1.6中datetimepicker标签的使用方法](http://mefly.javaeye.com/blog/393472)

datetimepicker标签在2.16已经不能像以前一样直接用了，须引入新的jar包，   
  
1.struts2-dojo-plugin-2.1.6.jar引入到自己的项目中。   
2.jsp中加入如下内容   
<%@ page language="java" pageEncoding="UTF-8"%>   
<%@ taglib prefix="s" uri="/ struts-tags" %>   
<%@ taglib prefix="sx" uri="/struts-dojo-tags" %>   
3. <head>中加入下列   
  
<head>   
    <s:head theme="xhtml"/>   
    <sx:head parseContent="true"/>         
</head>   
4.页面中使用   
<sx:datetimepicker name="birthday" lable="请求时间" displayFormat="yy-MM-dd" />

# 没有解决的问题