



珍爱网
zhenai.com
珍爱红娘
专业相亲

还单身?
你的机会来了

我要找: 女士 > 年龄: 22 > 28 > 立即搜索
地区: 广东 > 深圳 > ☒ 有照片
免费注册>>
QQ登录

论坛首页 → Java企业应用论坛 → 最新SpringMVC + spring3.1.1 + hibernate4.1.0 集成及常见问题总结

全部 Hibernate Spring Struts iBATIS 企业应用 Lucene SOA Java综合 设计模式 Tomcat OO JBoss

主题: 最新SpringMVC + spring3.1.1 + hibernate4.1.0 集成及常见问题总结 该帖已经被评为精华帖

作者

jinnianshilongnia

等级:

性别:

文章: 422

积分: 1310

我现在离线

正文

发表时间: 2012-02-26 最后修改: 2012-03-01

< >

 猎头职位: 上海: 【上海】 外资企业高新诚聘web开发工程师

Spring Hibernate 企业应用 SSH

源码下载地址

一 开发环境

1、动态web工程

2、部分依赖

java代码:

Java代码

1. hibernate-release-4.1.0.Final.zip

2. hibernate-validator-4.2.0.Final.jar

3. spring-framework-3.1.1.RELEASE-with-docs.zip

4. proxool-0.9.1.jar

5. log4j 1.2.16

6. slf4j -1.6.1

7. mysql-connector-java-5.1.10.jar

8. hamcrest 1.3.0RC2

9. ehcache 2.4.3

3、为了方便学习，暂没有使用maven构建工程

二 工程主要包括内容

1、springMVC + spring3.1.1 + hibernate4.1.0集成

2、通用DAO层 和 Service层

3、二级缓存 Ehcache

4、REST风格的表现层

5、通用分页（两个版本）

5.1、首页 上一页,下一页 尾页 跳转

相关文章:

▪ [\[petclinic专题\]查询没问题新增时报错、附发布过程](#)

▪ [Hibernate更新oracle数据库出现无法更新汉字的问题](#)

▪ [深入osworkflow的function](#)

推荐群组: JBPM @net

[更多相关推荐](#)

- 5.2、上一页 1 2 3 4 5 下一页
- 6、数据库连接池采用proxool
- 7、spring集成测试
- 8、表现层的 java validator框架验证（采用hibernate-validator-4.2.0实现）
- 9、视图采用JSP，并进行组件化分离

三 TODO LIST 将本项目做成脚手架方便以后新项目查询

- 1、Service层进行AOP缓存（缓存使用Memcached实现）
- 2、单元测试（把常见的桩测试、伪实现、模拟对象演示一遍 区别集成测试）
- 3、监控功能
- 后台查询hibernate二级缓存 hit/miss率功能
- 后台查询当前服务器状态功能（如 线程信息、服务器相关信息）
- 4、spring RPC功能
- 5、spring集成 quartz 进行任务调度
- 6、spring集成 java mail进行邮件发送
- 7、DAO层将各种常用框架集成进来（方便查询）
- 8、把工作中经常用的东西 融合进去，作为脚手架，方便以后查询

四 集成重点及常见问题

- 1、spring-config.xml 配置文件：
 - 1.1、该配置文件只加载除表现层之外的所有bean，因此需要如下配置：

java代码：

Java 代码



```
1. <context:component-scan base-package="cn.javass">
2.     <context:exclude-filter type="annotation" expression="org.springframework.stereotype.Controller"/>
3. </context:component-scan>
```

通过exclude-filter 把所有 @Controller注解的表现层控制器组件排除

- 1.2、国际化消息文件配置

java代码：

Java 代码



```
1. <!-- 国际化的消息资源文件 -->
2. <bean id="messageSource" class="org.springframework.context.support.ReloadableResourceBundleMessageSource">
3.     <property name="basenames">
4.         <list>
5.             <!-- 在web环境中一定要定位到classpath 否则默认到当前web应用下找 -->
6.             <value>classpath:messages</value>
7.         </list>
8.     </property>
9.     <property name="defaultEncoding" value="UTF-8"/>
10.    <property name="cacheSeconds" value="60"/>
11. </bean>
```

此处basenames内一定是 classpath:messages，如果你写出“messages”，将会到你的web应用的根下找 即你的messages.properties一定在 web应用/messages.propertis。

- 1.3、hibernate的sessionFactory配置 需要使用org.springframework.orm.hibernate4.LocalSessionFactoryBean，其他都是类

似的，具体看源代码。

1. 4、<aop:aspectj-autoproxy expose-proxy="true"/> 实现@AspectJ注解的，默认使用AnnotationAwareAspectJAutoProxyCreator进行AOP代理，它是BeanPostProcessor的子类，在容器启动时Bean初始化开始和结束时调用进行AOP代理的创建，因此只对当容器启动时有效，使用时注意此处。

1.5、声明式容器管理事务

建议使用声明式容器管理事务，而不建议使用注解容器管理事务（虽然简单），但太分布式了，采用声明式容器管理事务一般只对service层进行处理。

java代码:

Java代码 

```
1. <tx:advice id="txAdvice" transaction-manager="txManager">
2.   <tx:attributes>
3.     <tx:method name="save*" propagation="REQUIRED" />
4.     <tx:method name="add*" propagation="REQUIRED" />
5.     <tx:method name="create*" propagation="REQUIRED" />
6.     <tx:method name="insert*" propagation="REQUIRED" />
7.     <tx:method name="update*" propagation="REQUIRED" />
8.     <tx:method name="merge*" propagation="REQUIRED" />
9.     <tx:method name="del*" propagation="REQUIRED" />
10.    <tx:method name="remove*" propagation="REQUIRED" />
11.    <tx:method name="put*" propagation="REQUIRED" />
12.    <tx:method name="use*" propagation="REQUIRED"/>
13.    <!--hibernate4必须配置为开启事务 否则 getCurrentSession()获取不到-->
14.    <tx:method name="get*" propagation="REQUIRED" read-only="true" />
15.    <tx:method name="count*" propagation="REQUIRED" read-only="true" />
16.    <tx:method name="find*" propagation="REQUIRED" read-only="true" />
17.    <tx:method name="list*" propagation="REQUIRED" read-only="true" />
18.    <tx:method name="*" read-only="true" />
19.  </tx:attributes>
20. </tx:advice>
21. <aop:config expose-proxy="true">
22.   <!-- 只对业务逻辑层实施事务 -->
23.   <aop:pointcut id="txPointcut" expression="execution(* cn.javass..service..*(..))" />
24.   <aop:advisor advice-ref="txAdvice" pointcut-ref="txPointcut"/>
25. </aop:config>
```

此处一定注意 使用 hibernate4，在不使用OpenSessionInView模式时，在使用getCurrentSession()时会有如下问题：


当有一个方法list 传播行为为Supports，当在另一个方法getPage()（无事务）调用list方法时会抛出org.hibernate.HibernateException: No Session found for current thread 异常。
这是因为getCurrentSession()在没有session的情况下不会自动创建一个，不知道这是不是Spring3.1实现的bug，欢迎大家讨论下。

因此最好的解决方案是使用REQUIRED的传播行为。请看最后的分析

二、spring-servlet.xml:

2.1、表现层配置文件，只应加装表现层Bean，否则可能引起问题。

java代码:

Java代码 

```
1. <!-- 开启controller注解支持 -->
2. <!-- 注: 如果base-package=cn.javass 则注解事务不起作用 -->
```


```
3. <context:component-scan base-package="cn.javass.demo.web.controller">
4.     <context:include-filter type="annotation" expression="org.springframework.stereotype.Controller"/>
5. </context:component-scan>
```

此处只应该加载表现层组件，如果此处还加载dao层或服务层的bean会将之前容器加载的替换掉，而且此处不会进行AOP织入，所以会造成AOP失效问题（如事务不起作用），再回头看我们的1.4讨论的。

2.2、<mvc:view-controller path="/" view-name="forward:/index"/> 表示当访问主页时自动转发到index控制器。

2.3、静态资源映射

java代码:

Java代码 


```
1. <!-- 当在web.xml 中 DispatcherServlet使用 <url-pattern></url-pattern> 映射时，能映射静态资源 -->
2. <mvc:default-servlet-handler/>
3. <!-- 静态资源映射 -->
4. <mvc:resources mapping="/images/**" location="/WEB-INF/images/" />
5. <mvc:resources mapping="/css/**" location="/WEB-INF/css/" />
6. <mvc:resources mapping="/js/**" location="/WEB-INF/js/" />
```

以上是配置文件部分，接下来来看具体代码。

三、通用DAO层Hibernate4实现

为了减少各模块实现的代码量，实际工作时都会有通用DAO层实现，以下是部分核心代码：

java代码:

Java代码 

```
1. public abstract class BaseHibernateDao<M extends java.io.Serializable, PK extends java.io.Serializable>
2. implements IBaseDao<M, PK> {
3.
4.     protected static final Logger LOGGER = LoggerFactory.getLogger(BaseHibernateDao.class);
5.
6.     private final Class<M> entityClass;
7.     private final String HQL_LIST_ALL;
8.     private final String HQL_COUNT_ALL;
9.     private final String HQL_OPTIMIZE_PRE_LIST_ALL;
10.    private final String HQL_OPTIMIZE_NEXT_LIST_ALL;
11.    private String pkName = null;
12.
13.    @SuppressWarnings("unchecked")
14.    public BaseHibernateDao() {
15.        this.entityClass = (Class<M>) ((ParameterizedType) getClass().getGenericSuperclass()).getActualTypeArguments()[0];
16.        Field[] fields = this.entityClass.getDeclaredFields();
17.        for(Field f : fields) {
18.            if(f.isAnnotationPresent(Id.class)) {
19.                this.pkName = f.getName();
20.            }
21.        }
22.
23.        Assert.notNull(pkName);
24.        //TODO @Entity name not null
25.        HQL_LIST_ALL = "from " + this.entityClass.getSimpleName() + " order by " + pkName + " desc";
26.        HQL_OPTIMIZE_PRE_LIST_ALL = "from " + this.entityClass.getSimpleName() + " where " + pkName + " > ? order by " + pkName + " asc";
27.        HQL_OPTIMIZE_NEXT_LIST_ALL = "from " + this.entityClass.getSimpleName() + " where " + pkName + "
```

```
27.         " < ? order by " + pkName + " desc";
28.     }
29.
30.     @Autowired
31.     @Qualifier("sessionFactory")
32.     private SessionFactory sessionFactory;
33.
34.     public Session getSession() {
35.         //事务必须是开启的, 否则获取不到
36.         return sessionFactory.getCurrentSession();
37.     }
38.     .....
39. }
```

Spring3.1集成Hibernate4不再需要HibernateDaoSupport和HibernateTemplate了, 直接使用原生API即可。

四、通用Service层代码 此处省略, 看源代码, 有了通用代码后CURD就不用再写了。

java代码:

Java代码



```
1.  @Service("UserService")
2.  public class UserServiceImpl extends BaseService<UserModel, Integer> implements UserService {
3.
4.      private static final Logger LOGGER = LoggerFactory.getLogger(UserServiceImpl.class);
5.
6.      private UserDao userDao;
7.
8.      @Autowired
9.      @Qualifier("UserDao")
10.     @Override
11.     public void setBaseDao(IBaseDao<UserModel, Integer> userDao) {
12.         this.baseDao = userDao;
13.         this.userDao = (UserDao) userDao;
14.     }
15.
16.
17.
18.     @Override
19.     public Page<UserModel> query(int pn, int pageSize, UserQueryModel command) {
20.         return PageUtil.getPage(userDao.countQuery(command), pn, userDao.query(pn, pageSize, command), p
ageSize);
21.     }
22. }
23. }
```

五、表现层 Controller实现

采用SpringMVC支持的REST风格实现, 具体看代码, 此处我们使用了java Validator框架 来进行 表现层数据验证

在Model实现上加验证注解

java代码:

Java代码




```
1.  @Pattern(regexp = "[A-Za-z0-9]{5,20}", message = "{username.illegal}") //java validator验证 (用户名字母数字组成, 长度为5-10)
```

```
2. private String username;
3.
4. @NotEmpty(message = "{email.illegal}")
5. @Email(message = "{email.illegal}") // 错误消息会自动到MessageSource中查找
6. private String email;
7.
8. @Pattern(regexp = "[A-Za-z0-9]{5,20}", message = "{password.illegal}")
9. private String password;
10.
11. @DateFormat( message="{register.date.error}")// 自定义的验证器
12. private Date registerDate;
```

在Controller中相应方法的需要验证的参数上加@Valid即可

java代码:


Java代码 

```
1. @RequestMapping(value = "/user/add", method = {RequestMethod.POST})
2. public String add(Model model, @ModelAttribute("command") @Valid UserModel command, BindingResult result)
   )
```

六、Spring集成测试

使用Spring集成测试能很方便的进行Bean的测试，而且使用@TransactionConfiguration(transactionManager = "txManager", defaultRollback = true)能自动回滚事务，清理测试前后状态。

java代码:

Java代码 


```
1. @RunWith(SpringJUnit4ClassRunner.class)
2. @ContextConfiguration(locations = {"classpath:spring-config.xml"})
3. @Transactional
4. @TransactionConfiguration(transactionManager = "txManager", defaultRollback = true)
5. public class UserServiceTest {
6.
7.     AtomicInteger counter = new AtomicInteger();
8.
9.     @Autowired
10.    private UserService userService;
11.    .....
12. }
```

其他部分请直接看源码，欢迎大家讨论。

补充spring3.1.1源代码分析当 传播行为为 Support时报 org.hibernate.HibernateException: No Session found for current thread 异常:


spring3.1开始 不提供（没有这个东西了）Hibernate4的 DaoSupport和Template，，而是直接使用原生的Hibernate4 API

如在 Hibernate3中 HibernateTemplate中有如下代码

Java代码 


```
1.  protected Session getSession() {
2.      if (isAlwaysUseNewSession()) {
3.          return SessionFactoryUtils.getNewSession(getSessionFactory(), getEntityInterceptor());
4.      }
5.      else if (isAllowCreate()) { //默认是true, 也就是即使你的传播行为是Supports也一定会有session存在的
6.          return SessionFactoryUtils.getSession(
7.              getSessionFactory(), getEntityInterceptor(), getJdbcExceptionTranslator());
8.      }
9.      else if (SessionFactoryUtils.hasTransactionalSession(getSessionFactory())) {
10.         return SessionFactoryUtils.getSession(getSessionFactory(), false);
11.     }
12.     else {
13.         try {
14.             return getSessionFactory().getCurrentSession();
15.         }
16.         catch (HibernateException ex) {
17.             throw new DataAccessResourceFailureException("Could not obtain current Hibernate Session",
18.                 ex);
19.         }
20.     }
```

但我们使用的是Hibernate4原生API，使用SpringSessionContext获取session，而这个isAllowCreate选项默认为false

Java代码 

```
1.  /**
2.   * Retrieve the Spring-managed Session for the current thread, if any.
3.   */
4.  public Session currentSession() throws HibernateException {
5.      try {
6.          return (org.hibernate.classic.Session) SessionFactoryUtils.doGetSession(this.sessionFactory, false)
7.          ; //最后的false即是
8.      }
9.      catch (IllegalStateException ex) {
10.         throw new HibernateException(ex.getMessage());
11.     }
```

SessionFactoryUtils类

Java代码 

```
1.  public static Session doGetSession(SessionFactory sessionFactory, boolean allowCreate)
2.      throws HibernateException, IllegalStateException {
3.
4.      return doGetSession(sessionFactory, null, null, allowCreate);
5.  }
```

可否认为这是集成Hibernate4的bug，或者采用OpenSessionInView模式解决或者传播行为最低为Required。

原创内容，转载请注明私塾在线【<http://sishuok.com/forum/blogPost/list/2625.html>】

声明：ITeye 文章版权属于作者，受法律保护。没有作者书面许可不得转载。

推荐链接

- 想进外企，出国，跳槽，找工作？英语不好,快来充电吧
- Java讲师与Android讲师通缉令--悬赏2000元！



稳定的jsp空间 3..
数据库不限大小,独享TOMCAT 试用7



皇和教育网络培训学校
专业的VCP培训,VCAP培训,VCDX

返回顶楼

[主页](#) [资料](#) [短信](#) [留言](#) [关注](#)

巴巴米
等级: 初级会员



性别:
文章: 18
积分: 40
来自: 长沙
[我现在离线](#)

发表时间: 2012-02-26

hibernate都4.1了啊...

[查看图片附件](#)

返回顶楼

[主页](#) [资料](#) [短信](#) [留言](#) [关注](#) 回帖地址



0



0

请登录投票

jinnianshilongnia
等级:



性别:
文章: 422
积分: 1310
[我现在离线](#)

发表时间: 2012-02-26

巴巴米 写道

hibernate都4.1了啊...

是。hibernate4发布都快一年了

返回顶楼

[主页](#) [资料](#) [短信](#) [留言](#) [关注](#) 回帖地址



0



0

请登录投票


finallygo
等级: 初级会员



性别:
文章: 280

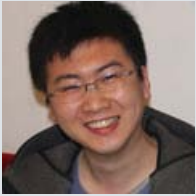
发表时间: 2012-02-26

个人怀疑,"不知道这是不是Spring3.1实现的bug "和你使用原生态session有关系,你试试使用hibernateTemplate看看有没有这个问题

积分: 30
来自: 厦门-->北京
 我现在离线

[返回顶楼](#) [主页](#) [资料](#) [短信](#) [留言](#) [关注](#) [回帖地址](#)  0  0 请登录投票

jinnianshilongnia
等级:  



性别: 
文章: 422
积分: 1310
 我现在离线

发表时间: 2012-02-26

finallygo 写道

个人怀疑,"不知道这是不是Spring3.1实现的bug "和你使用原生态session有关系,你试试使用hibernateTemplate看看有没有这个问题

spring3.1+hibernate4集成时 不再用 DaoSupport和 Template了, , 直接使用原生API。

[返回顶楼](#) [主页](#) [资料](#) [短信](#) [留言](#) [关注](#) [回帖地址](#)  0  0 请登录投票

finallygo
等级: 初级会员



性别: 
文章: 280
积分: 30
来自: 厦门-->北京
 我现在离线

发表时间: 2012-02-26

jinnianshilongnian 写道


finallygo 写道

个人怀疑,"不知道这是不是Spring3.1实现的bug "和你使用原生态session有关系,你试试使用hibernateTemplate看看有没有这个问题

spring3.1+hibernate4集成时 不再用 DaoSupport和 Template了, , 直接使用原生API。

嗯,我知道可以,我的意思是你试试如果还是使用hibernateTemplate的话,看看有没有这个问题

[返回顶楼](#) [主页](#) [资料](#) [短信](#) [留言](#) [关注](#) [回帖地址](#)  0  0 请登录投票

jinnianshilongnia
等级:  



性别: 
文章: 422
积分: 1310
 我现在离线

发表时间: 2012-02-26 最后修改: 2012-02-26

finallygo 写道

jinnianshilongnian 写道

finallygo 写道

个人怀疑,"不知道这是不是Spring3.1实现的bug "和你使用原生态session有关系,你试试使用hibernateTemplate看看有没有这个问题

spring3.1+hibernate4集成时 不再用 DaoSupport和 Template了，，直接使用原生API。

嗯,我知道可以,我的意思是你试试如果还是使用hibernateTemplate的话,看看有没有这个问题

spring3.1开始 不提供（没有这个东西了）Hibernate4的 DaoSupport和Template，，而是直接使用原生的Hibernate4 API

如在 Hibernate3中 HibernateTemplate中有如下代码

Java代码



```
1.  protected Session getSession() {
2.             if (isAlwaysUseNewSession()) {
3.                 return SessionFactoryUtils.getNewSession(getSessionFactory(), getEntityInterceptor
4.             );
5.             }
6.             else if (isAllowCreate()) { //默认是true, 也就是即使你的传播行为是Supports也一定会有session存在
7.                 return SessionFactoryUtils.getSession(
8.                     getSessionFactory(), getEntityInterceptor(), getJdbcExceptionTran
9.                 slator());
10.            }
11.            else if (SessionFactoryUtils.hasTransactionalSession(getSessionFactory())) {
12.                return SessionFactoryUtils.getSession(getSessionFactory(), false);
13.            }
14.            else {
15.                try {
16.                    return getSessionFactory().getCurrentSession();
17.                }
18.                catch (HibernateException ex) {
19.                    throw new DataAccessResourceFailureException("Could not obtain current Hi
20.                bernate Session", ex);
21.            }
22.        }
```

但我们使用的是Hibernate4原生API，使用SpringSessionContext获取session，而这个isAllowCreate选项默认为false

Java代码



```
1.  /**
2.   * Retrieve the Spring-managed Session for the current thread, if any.
3.   */
4.  public Session currentSession() throws HibernateException {
5.      try {
6.          return (org.hibernate.classic.Session) SessionFactoryUtils.doGetSession(this.sessionFactory
7.      , false); //最后的false即是
8.      }
9.      catch (IllegalStateException ex) {
10.         throw new HibernateException(ex.getMessage());
11.     }
12. }
```

SessionFactoryUtils类

Java代码

```
1. public static Session doGetSession(SessionFactory sessionFactory, boolean allowCreate)
2.     throws HibernateException, IllegalStateException {
3.
4.     return doGetSession(sessionFactory, null, null, allowCreate);
5. }
```

可否认为这是集成Hibernate4的bug。可以使用openSessionInView模式解决，或直接Required。

返回顶楼

主页 资料 短信 留言 关注 回帖地址

0 0 请登录投票

finallygo
等级: 初级会员



性别:
文章: 280
积分: 30
来自: 厦门-->北京
 我现在离线

发表时间: 2012-02-26

可以使用吧,只是不推荐了,请参考如下链接:
<http://static.springsource.org/spring/docs/3.1.x/spring-framework-reference/html/classic-spring.html#classic-spring-orm>

返回顶楼

主页 资料 短信 留言 关注 回帖地址

0 0 请登录投票

jinnianshilongnia
等级:



性别:
文章: 422
积分: 1310
 我现在离线

发表时间: 2012-02-26

finallygo 写道

可以使用吧,只是不推荐了,请参考如下链接:
<http://static.springsource.org/spring/docs/3.1.x/spring-framework-reference/html/classic-spring.html#classic-spring-orm>

你引用的是说Hibernate3， 我们现在讨论的是Spring3.1集成Hibernate4

返回顶楼

主页 资料 短信 留言 关注 回帖地址

0 0 请登录投票

finallygo
等级: 初级会员



发表时间: 2012-02-26

finallygo 写道

可以使用吧,只是不推荐了,请参考如下链接:
<http://static.springsource.org/spring/docs/3.1.x/spring-framework-reference/html/classic-spring.html#classic-spring-orm>

性别:

文章: 280

积分: 30

来自: 厦门-->北京

我现在离线

好吧,是我错了,这个是hibernate3的

返回顶楼

主页

资料

短信

留言

关注

回帖地址

0

0 请登录投票

发表回复

« 上一页

1

2

3

...

12

13

下一页 »

[论坛首页](#) → [Java企业应用版](#)

跳转论坛:

Java

- [北京: 中油瑞飞/richfit诚聘JAVA开发工程师](#)
- [北京: 美团诚聘研发工程师 \(Java\)](#)
- [上海: 用友汽车软件诚聘中级java程序员 \(工作地点:](#)
- [湖北: 用友汽车软件诚聘项目经理 \(武汉\)](#)
- [广东: 用友汽车软件诚聘中级JAVA工程师 工作地点:](#)
- [北京: 中国人寿财险诚聘英才项目管理岗](#)

[广告服务](#) | [ITeye黑板报](#) | [联系我们](#) | [友情链接](#)

© 2003-2011 ITeye.com. [[京ICP证110151号](#) [京公网安备110105010620](#)]
百联优力(北京)投资有限公司 版权所有
上海炯耐计算机软件有限公司 提供商务支持