灯火阑珊处

永久域名 http://shutiao2008.javaeye.com

shutiao2008
浏览: 23891 次
性别:
来自: 杭州
我现在离线
详细资料 留言簿

搜索本博客

最近访客 >>更多访客

pengcqu

westice

lincs

disankyo

博客分类

- 全部博客 (68)
- php学习 (1)
- 转载 (9)
- rubyrails (2)
- linux (12)
- db (17)
- erlang (1)
- spring (2)
- hadoop (4)
- java (8)

优化MySQL数据库性能的八种方法 | Hibernate3.3的Annotation的实现

2009-03-25

基于Annotation的Hibernate3.3+Spring2.5整合开发

关键字: annotation, hibernate, spring

现在，我们加入Spring的支持：把spring-framework-2.5.5\dist中的spirng.jar引进我们项目的lib目录来，还要添加aspectjweaver.jar包，以支持切面编程。

必要的配置文件还是要的：

applicationContext-common.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:context="http://www.springframework.org/schema/context"
   xmlns:aop="http://www.springframework.org/schema/aop"
   xmlns:tx="http://www.springframework.org/schema/tx"
   xsi:schemaLocation="
         http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-2.5.xsd
         http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-2.5.xsd
         http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-2.5.xsd
         http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-2.5.xsd">

   <!-- 配置SessionFactory,由Spring容器来管理Hibernate -->
   <!-- 非Annotation时,使用org.springframework.orm.hibernate3.LocalSessionFactoryBean,
      它注入实体类的方式是setMappingResources(),而Hibernate Annotation所用的映射方式
      不是mapping resource,而是mapping class,这就要用到LocalSessionFactoryBean的子类
      AnnotationSessionFactoryBean了.因为AnnotationSessionFactoryBean它支持实体的注入
      方式setAnnotatedClasses,即对应Hibernate中的mapping class.参见这两个类的源代码. -->
   <bean id="sessionFactory"
      class="org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean">
      <property name="configLocation">
         <value>classpath:hibernate.cfg.xml</value>
      </property>
   </bean>
   <!-- 配置事务管理器 -->
   <bean id="transactionManager"
      class="org.springframework.orm.hibernate3.HibernateTransactionManager">
      <property name="sessionFactory">
         <ref bean="sessionFactory" />
      </property>
   </bean>

   <!-- 配置事务的传播特性 -->
   <tx:advice id="txAdvice" transaction-manager="transactionManager">
      <tx:attributes>
         <tx:method name="save*" propagation="REQUIRED" />
         <tx:method name="update*" propagation="REQUIRED" />
         <tx:method name="delete*" propagation="REQUIRED" />
         <tx:method name="*" read-only="true" />
      </tx:attributes>
   </tx:advice>
```

apache & iis (2)
- lua (2)
- python (2)
- web开发 (4)
- felix (3)

其他分类
- 我的收藏 (33)
- 我的论坛主题贴 (0)
- 我的所有论坛贴 (1)
- 我的精华良好贴 (0)

最近加入圈子
- Python

存档
- 2010-06 (3)
- 2010-05 (15)
- 2010-02 (1)
- 更多存档...

最新评论
- hadoop put异常
  我也碰到这样的问题了，我是单机的伪分布式，你的意思是start后多等一会就可以了？ ...
  -- by HRoger

评论排行榜
- mysql sphinx存储引擎安装
- 安装Sphinx存储引擎
- Sphinx使用入门
- python中文相关
- GCC入门

RSS
Add to Google
订阅到 鲜果
[什么是RSS?]

```
    <!-- 那些类的哪些方法参与事务 -->
    <aop:config>
        <aop:pointcut id="allServiceMethod" expression="execution(* com.rong.dao.*.*.*(..))" />
        <aop:advisor pointcut-ref="allServiceMethod" advice-ref="txAdvice" />
    </aop:config>

    <!-- 使Spring关注Annotation -->
    <context:annotation-config/>

    <!-- 让Spring通过自动扫描来查询和管理Bean -->
    <context:component-scan base-package="com.rong"/>

    <!--
    <bean id="userDao" class="com.rong.dao.UserDaoBean">
        <property name="sessionFactory" ref="sessionFactory"/>
    </bean>
    <bean id="userService" class="com.rong.service.UserServiceBean">
        <property name="userDao" ref="userDao"/>
    </bean>
     -->

</beans>
```

关键的两点：

```
<!-- 使Spring关注Annotation -->
<context:annotation-config/>

<!-- 让Spring通过自动扫描来查询和管理Bean -->
<context:component-scan base-package="com.rong"/>
```

这样配置之后，就省去了上面注释掉的DAO层和Service层等配置代码。是不是很方便呢。
关于这一部分的XML代码，我们下面还会作解释。

来开发我们的DAO层吧,接口如下：

```
package com.rong.dao;
import java.util.List;
import com.rong.entity.User;
public interface UserDao {

    public void save(User user);

    public void delete(int id);

    public void update(User user);

    public List<User> query();

    public User get(int id);
}
```

DAO层的实现类：

```
package com.rong.dao;
import java.util.List;
import org.springframework.stereotype.Repository;
import com.rong.entity.User;
@Repository("userDao")      //声明此类为数据持久层的类
public class UserDaoBean extends MyHibernateDaoSupport implements UserDao {

    public void save(User user){
        super.getHibernateTemplate().save(user);
    }

    public void delete(int id){
        super.getHibernateTemplate().delete(super.getHibernateTemplate().load(User.class, id));
    }
```

```
    public void update(User user){
        super.getHibernateTemplate().update(user);
    }

    @SuppressWarnings("unchecked")
    public List<User> query(){
        return super.getHibernateTemplate().find("from User");
    }

    public User get(int id){
        return (User)super.getHibernateTemplate().get("from User", id);
    }
}
```

大家可以看到，我们这里继承的不是HibernateDaoSupport,而是我自己编写的一个类MyHibernateDaoSupport。其代码如下：

```
package com.rong.dao;
import javax.annotation.Resource;
import org.hibernate.SessionFactory;
import org.springframework.orm.hibernate3.support.HibernateDaoSupport;
public class MyHibernateDaoSupport extends HibernateDaoSupport {

    @Resource(name="sessionFactory")    //为父类HibernateDaoSupport注入sessionFactory的值
    public void setSuperSessionFactory(SessionFactory sessionFactory){
        super.setSessionFactory(sessionFactory);
    }
}
```

我们之所以要改写HibernateDaoSupport，是因我为，我们要为DAO层的类注入SessionFactory这个属性。以后，我们开发的DAO类，就可以直接重用这个MyHibernateDaoSupport了。其实，这样做是相当于配置文件方式的代码：

```
<bean id="userDao" class="com.rong.dao.UserDaoBean">
    <property name="sessionFactory" ref="sessionFactory"/>
</bean>
```

我们既然要用annotation代替XML文件的，就要让它也能像原来那样使用sessionFactory,故为MyHibernateDaoSupport注入SessionFactory。子类继承这个类时，也继承其Annotation。这样，我们就可以实现SessionFactory的注入了。

到现在，我们再回过头来看applicationContext-common.xml中的

```
<bean id="sessionFactory"
      class="org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean">
      <property name="configLocation">
        <value>classpath:hibernate.cfg.xml</value>
      </property>
 </bean>
```

我们平时开发Hibernate与Spring整合时，常常会用到org.springframework.orm.hibernate3.LocalSessionFactoryBean来提供SessionFactory，而我们这里却要改成org.springframework.orm.hibernate3.annotation.AnnotationSessionFactoryBean。其实是这样的，我们在Hibernate.cfg.xml中配置的实体类映射的方式如下：

```
<!--
<mapping resource="com/rong/entity/User.hbm.xml"/>
-->

<!-- 在Hibernate中注册User实体类,区别于上面注释掉的resource写法 -->
<mapping class="com.rong.entity.User"/>
```

要使Hibernate的实体类支持注解，去掉xxx.hbm.xml的文件，故我们所用的是mapping class方式，不是mapping resource的方法。然而，LocalSessionFactoryBean这个类，它采用的实体类映射方式是mapping resource，（详情可参见LocalSessionFactoryBean这个类的源代码）。如果我们在配置中仍然用这个类的话，Hibernate与Spring整合时，就会报错。而AnnotationSessionFactoryBean这个类在LocalSessionFactoryBean的基础上添加了mapping class方式实现实体类映射（详见AnnotationSessionFactoryBean类的源代码）。

我们再来看Service层的代码：（接口比较简单，节约篇幅就不列出了）

```
package com.rong.service;
import java.util.List;
import javax.annotation.Resource;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import com.rong.dao.UserDao;
import com.rong.entity.User;
@Service("userService")       //声明此类为业务逻辑层的类
public class UserServiceBean implements UserService {

    @Autowired
```

```
    private UserDao userDao;
    public void save(User user){
        userDao.save(user);
    }
}
```

我们用到的注解上面一般都作了注释，就不多叙。@Autowired和@Resource功能差不多，就是把对象注入，相当于<bean>配置的功能。

好，就开发到这样，是不是忘记了什么？记得要配置web.xml，部分代码

```
<!-- 修改Spring配置文件的路径 -->
<context-param>
    <param-name>contextConfigLocation</param-name>
    <param-value>classpath*:applicationContext-*.xml</param-value>
</context-param>

<!-- 配置Spring -->
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
```

其实，到现在为止，我们发觉我们的XML配置文件还是很多。其实，这样想想，上一阶段我们省去了xxx.hbm.xml这类的文件，这一阶段，我们少去了<bean id="" class=""><property name="" ref="">这样的配置项。而这些，正是我们项目开发中，大量使用的配置。而只要书写简单的Annotation注解，就可以省去这样，我们何乐而不用。而那些我们保留的XML配置文件（如：数据库连接，事务），这样是写死的，一个项目就写一次或复制过来用，我们保留它又何妨？

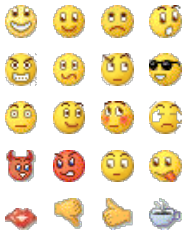
优化MySQL数据库性能的八种方法 | Hibernate3.3的Annotation的实现

18:37 | 浏览 (753) | 评论 (0) | 分类: java | 相关推荐

评论

发表评论

表情图标



B I U Quote Code List Img URL Flash Table

字体颜色: 标准 字体大小: 标准 对齐: 标准

提示：选择您需要装饰的文字，按上列按钮即可添加上相应的标签

您还没有登录，请登录后发表评论(快捷键 Alt+S / Ctrl+Enter) 提交