

Spring4+Hibernate4+Atomikos3.3多数据源事务管理 - Spring——Java程序员的春天

Spring3+后不再对JTOM提供支持，所以可以改用Atomikos管理多数据源事务。

Spring2.5+Hibernate3+JTOM参考：<http://hanqunfeng.iteye.com/blog/1554251>

Atomikos官网网站：<http://www.atomikos.com/>

一.pom.xml

Xml代码



二.applicationContext-atomikos.xml

1. <bean class="com.atomikos.jdbc.AtomikosDataSourceBean" init-method="init" destroy-method="close" id="dataSource01">
2. <!-- Set unique name for this DataSource -->
3. <property name="uniqueResourceName"><value>oracle</value></property>
4. <!-- Set XADatasource class name -->
5. <property name="xaDataSourceClassName" value="oracle.jdbc.xa.client.OracleXADataSource" />
6. <property name="xaProperties">
7. <props>
8. <prop key="user">\${jdbc.username01}</prop>
9. <prop key="password">\${jdbc.password01}</prop>
10. <prop key="URL">\${jdbc.url01}</prop>
11. </props>
12. </property>
13. <!-- set properties for datasource connection pool -->
14. <property name="poolSize" value="3" />
15. <!-- timeout after 20000 seconds -->
16. <property name="reapTimeout"><value>20000</value></property>

```
17. </bean>
18. <bean id="sessionFactory01"
19.     class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">
20.     <property name="dataSource" ref="dataSource01" />
21.     <property name="packagesToScan">
22.         <list>
23.             <value>com.cpframework.function.**.model.oracle</value>
24.         </list>
25.     </property>
26.     <property name="hibernateProperties">
27.         <props>
28.             <prop key="hibernate.dialect">
29.                 org.hibernate.dialect.Oracle10gDialect
30.             </prop>
31.             <prop key="hibernate.show_sql">true</prop>
32.             <prop key="hibernate.current_session_context_class">jta</prop>
33.             <prop key="hibernate.transaction.factory_class">org.hibernate.transaction.JT
ATransactionFactory</prop>
34.         </props>
35.     </property>
36. </bean>
37. <!-- oracleDAO -->
38. <bean id="hibernateDAO01" class="org.cpframework.dao.hibernate4.CP_Hibernate4
DAOImpl">
39.     <property name="sessionFactory" ref="sessionFactory01"></property>
40. </bean>
41. <bean class="com.atomikos.jdbc.AtomikosDataSourceBean" init-
method="init" destroy-method="close" id="dataSource02">
42.     <!-- Set unique name for this DataSource -->
43.     <property name="uniqueResourceName"><value>mysql</value></property>
44.     <!-- Set XADataSource class name -->
45.     <property name="xaDataSourceClassName" value="com.mysql.jdbc.jdbc2.optional
.MysqlXADataSource" />
46.     <property name="xaProperties">
47.         <props>
48.             <prop key="user">${jdbc.username02}</prop>
```

```
49.         <prop key="password">${jdbc.password02}</prop>
50.         <prop key="URL">${jdbc.url02}</prop>
51.     </props>
52. </property>
53. <!-- set properties for datasource connection pool -->
54. <property name="poolSize" value="3" />
55. <property name="reapTimeout"><value>20000</value></property>
56. </bean>
57. <bean id="sessionFactory02"
58.     class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">
59.     <property name="dataSource" ref="dataSource02" />
60.     <property name="packagesToScan">
61.         <list>
62.             <value>com.cpframework.function.**.model.mysql</value>
63.         </list>
64.     </property>
65.     <property name="hibernateProperties">
66.         <props>
67.             <prop key="hibernate.dialect">
68.                 org.hibernate.dialect.MySQLDialect
69.             </prop>
70.             <prop key="hibernate.show_sql">true</prop>
71.             <prop key="hibernate.current_session_context_class">jta</prop>
72.             <prop key="hibernate.transaction.factory_class">org.hibernate.transaction.JT
ATransactionFactory</prop>
73.         </props>
74.     </property>
75. </bean>
76. <!-- mysqlDAO -->
77. <bean id="hibernateDAO02" class="org.cpframework.dao.hibernate4.CP_Hibernate4
DAOImpl">
78.     <property name="sessionFactory" ref="sessionFactory02"></property>
79. </bean>
80. <!-- atomikos事务管理器 -->
81. <bean id="atomikosTransactionManager" class="com.atomikos.icatch.jta.UserTransa
ctionManager"
```

```
82.     init-method="init" destroy-method="close">
83.     <description>UserTransactionManager</description>
84.     <property name="forceShutdown">
85.         <value>true</value>
86.     </property>
87. </bean>
88. <bean id="atomikosUserTransaction" class="com.atomikos.icatch.jta.UserTransactio
    nImp">
89.     <property name="transactionTimeout" value="300" />
90. </bean>
91. <!-- spring 事务管理器 -->
92. <bean id="transactionManager"
93.     class="org.springframework.transaction.jta.JtaTransactionManager">
94.     <property name="transactionManager" ref="atomikosTransactionManager"/>
95.     <property name="userTransaction" ref="atomikosUserTransaction" />
96.     <property name="allowCustomIsolationLevels" value="true"/>
97. </bean>
98. <aop:config proxy-target-class="true">
99.     <aop:pointcut id="servicePoint"
100.         expression="execution (* com.cpframework.function.*.service.*.*Service*.*
    (..))" />
101.     <aop:advisor pointcut-ref="servicePoint" id="txAdvisor"
102.         advice-ref="defaultTxAdvice" />
103. </aop:config>
104. <tx:advice id="defaultTxAdvice" transaction-manager="transactionManager">
105.     <tx:attributes>
106.         <tx:method name="find*" read-only="true" />
107.         <tx:method name="get*" read-only="true" />
108.         <tx:method name="query*" read-only="true" />
109.         <tx:method name="load*" read-only="true" />
110.         <tx:method name="add*" propagation="REQUIRED" rollback-
    for="java.lang.Exception" />
111.         <tx:method name="create*" propagation="REQUIRED"
112.             rollback-for="java.lang.Exception" />
113.         <tx:method name="save*" propagation="REQUIRED"
114.             rollback-for="java.lang.Exception" />
```

```
115.      <tx:method name="update*" propagation="REQUIRED"
116.          rollback-for="java.lang.Exception" />
117.      <tx:method name="modify*" propagation="REQUIRED"
118.          rollback-for="java.lang.Exception" />
119.      <tx:method name="delete*" propagation="REQUIRED"
120.          rollback-for="java.lang.Exception" />
121.      <tx:method name="remove*" propagation="REQUIRED"
122.          rollback-for="java.lang.Exception" />
123.      <tx:method name="apply*" propagation="REQUIRED"
124.          rollback-for="java.lang.Exception" />
125.      <tx:method name="handle*" propagation="REQUIRED"
126.          rollback-for="java.lang.Exception" />
127.      <tx:method name="do*" propagation="REQUIRED"
128.          rollback-for="java.lang.Exception" />
129.  </tx:attributes>
130. </tx:advice>
```

三.transactions.properties

将transactions.properties文件放到编译根路径下，可根据需要开启相关注解，参见附件。

四.CP_Hibernate4DAOImpl

需要注意两点：

- 1.session必须使用sessionFactory.openSession()的方式获得，不能使用sessionFactory.getCurrentSession()。
- 2.更新操作必须调用session.flush()方法。

例如：

Java代码

