## 1、<context:annotation-config> 和 <context:component-scan>的区别

<context:annotation-config> **是用于激活那些已经在spring容器里注册过的bean，也就是说在spring中添加好了bean，需要被resource**（无论是通过xml的方式还是通过package sanning的方式）上面的注解。

<context:component-scan>除了具有<context:annotation-config>的功能之外，<context:component-scan>还可以在指定的package下扫描以及注册javabean 。

## 2、事务管理器

|  |
| --- |
| <bean id=*"transactionManager"* class=*"org.springframework.jdbc.datasource.DataSourceTransactionManager"*>  <property name=*"dataSource"* ref=*"dataSource"* />  </bean> |

## 3、 <tx:annotation-driven 用于启动事物注解（加入事务管理器）

<tx:annotation-driven transaction-manager=*"transactionManager"* proxy-target-class=*"false"*/>

# 1、spring 开发环境选择

## 1、web.xml中配置选择哪种方式

|  |
| --- |
| <!-- 环境切换配置：production/develop/test -->  <context-param>  <param-name>spring.profiles.active</param-name>  <param-value>develop</param-value>  </context-param> |

## 2、spring 配置文件中添加两种数据源配置信息（这里是通过将数据源配置信息写到了properties文件进行，也可以在spring配置文件中写上多找了国内数据源）

<http://blog.csdn.net/hejingyuan6/article/details/51243985>

数据源定义为

**[html]** [view plain](http://blog.csdn.net/hejingyuan6/article/details/51243985) [copy](http://blog.csdn.net/hejingyuan6/article/details/51243985)

[print?](http://blog.csdn.net/hejingyuan6/article/details/51243985)

1. **<bean** id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"**>**
2. **<property** name="user" value="${jdbc.user}" **/>**
3. **<property** name="password" value="${jdbc.password}" **/>**
4. **<property** name="jdbcUrl" value="${jdbc.jdbcUrl}" **/>**
5. **<property** name="driverClass" value="${jdbc.driverClass}" **/>**
6. **<property** name="initialPoolSize" value="${c3p0.initialPoolSize}"**/>**
7. **<property** name="acquireIncrement" value="${c3p0.acquireIncrement}"**/>**
8. **<property** name="minPoolSize" value="${c3p0.minPoolSize}"**/>**
9. **<property** name="maxIdleTime" value="${c3p0.maxIdleTime}"**/>**
10. **<property** name="idleConnectionTestPeriod" value="${c3p0.idleConnectionTestPeriod}" **/>**
11. **<property** name="preferredTestQuery" value="${c3p0.preferredTestQuery}"**/>**
12. **</bean>**

classpath下外部资源文件有两个 settings-development.properties 和 settings-production.properties，分别是开发环境和生产环境的数据源配置参数，内容如下

settings-development.properties

**[html]** [view plain](http://blog.csdn.net/hejingyuan6/article/details/51243985) [copy](http://blog.csdn.net/hejingyuan6/article/details/51243985)

[print?](http://blog.csdn.net/hejingyuan6/article/details/51243985)

1. jdbc.user=root
2. jdbc.password=111111
3. jdbc.driverClass=com.mysql.jdbc.Driver
4. jdbc.jdbcUrl=jdbc:mysql://localhost:3306/xxx
5. c3p0.minPoolSize=5
6. c3p0.initialPoolSize=5
7. c3p0.acquireIncrement=5
8. c3p0.maxIdleTime=3600
9. c3p0.idleConnectionTestPeriod=3600
10. c3p0.preferredTestQuery=select 1

settings-production.properties

**[html]** [view plain](http://blog.csdn.net/hejingyuan6/article/details/51243985) [copy](http://blog.csdn.net/hejingyuan6/article/details/51243985)

[print?](http://blog.csdn.net/hejingyuan6/article/details/51243985)

1. jdbc.user=xxx
2. jdbc.password=xxxx
3. jdbc.driverClass=com.mysql.jdbc.Driver
4. jdbc.jdbcUrl=jdbc:mysql:///xxx
5. c3p0.minPoolSize=20
6. c3p0.initialPoolSize=20
7. c3p0.acquireIncrement=10
8. c3p0.maxIdleTime=3600
9. c3p0.idleConnectionTestPeriod=3600
10. c3p0.preferredTestQuery=select 1

#### 1. 定义 profile

现在就可以通过定义 profile 来将开发和生产环境的数据源配置分开，这里我们定义两个 profile，一个名称是 development，另一个名称是 production（配置到Spring的配置文件中即可）

**[html]** [view plain](http://blog.csdn.net/hejingyuan6/article/details/51243985) [copy](http://blog.csdn.net/hejingyuan6/article/details/51243985)

[print?](http://blog.csdn.net/hejingyuan6/article/details/51243985)

1. <!-- 开发环境配置文件 -->
2. **<beans** profile="development"**>**
3. **<context:property-placeholder** location="classpath:settings-development.properties"**/>**
4. **</beans>**
6. <!-- 生产环境配置文件 -->
7. **<beans** profile="production"**>**
8. **<context:property-placeholder** location="classpath:settings-production.properties"**/>**
9. **</beans>**

#### 2. 定义默认 profile

默认 profile 是指在没有任何 profile 被激活的情况下，默认 profile 内定义的内容将被使用，通常可以在 web.xml 中定义全局 servlet 上下文参数 spring.profiles.default 实现，代码如下

**[html]** [view plain](http://blog.csdn.net/hejingyuan6/article/details/51243985) [copy](http://blog.csdn.net/hejingyuan6/article/details/51243985)

[print?](http://blog.csdn.net/hejingyuan6/article/details/51243985)

1. <!-- 配置spring的默认profile -->
2. **<context-param>**
3. **<param-name>**spring.profiles.default**</param-name>**
4. **<param-value>**development**</param-value>**
5. **</context-param>**

当然此种方式也是使用默认profile的方式，即如果不指定激活哪个环境则使用默认方式加载文件

**[html]** [view plain](http://blog.csdn.net/hejingyuan6/article/details/51243985) [copy](http://blog.csdn.net/hejingyuan6/article/details/51243985)

[print?](http://blog.csdn.net/hejingyuan6/article/details/51243985)

1. **<beans** profile="default"**>**
2. **<context:property-placeholder** ignore-resource-not-found="true" location="classpath:application.properties"**/>**
3. **</beans>**

## 3、配置文件中写死 数据源信息

<http://blog.csdn.net/figo0423/article/details/54925993>

1. **<span** style="font-size:18px;"**>**  **<context-param>**
2. **<param-name>**spring.profiles.active**</param-name>**
3. **<param-value>**test**</param-value>**
4. **</context-param>**  **</span>**

             在spring xml配置文件这么配置

**[html]** [view plain](http://blog.csdn.net/figo0423/article/details/54925993) [copy](http://blog.csdn.net/figo0423/article/details/54925993)

1. **<span** style="font-size:18px;"**><?xml** version="1.0" encoding="UTF-8"**?>**
2. **<beans** xmlns="http://www.springframework.org/schema/beans"
3. xmlns:jee="http://www.springframework.org/schema/jee" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4. xmlns:jpa="http://www.springframework.org/schema/data/jpa" xmlns:tx="http://www.springframework.org/schema/tx"
5. xsi:schemaLocation="http://www.springframework.org/schema/beans
6. http://www.springframework.org/schema/beans/spring-beans-3.1.xsd
7. http://www.springframework.org/schema/data/jpa
8. http://www.springframework.org/schema/data/jpa/spring-jpa-1.1.xsd
9. http://www.springframework.org/schema/tx
10. http://www.springframework.org/schema/tx/spring-tx-3.1.xsd
11. http://www.springframework.org/schema/jee
12. http://www.springframework.org/schema/jee/spring-jee-3.1.xsd"**>**


16. **<bean** id="propertyConfigurer"
17. class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"**>**
18. **<property** name="location" value="classpath:database.properties" **/>**
19. **</bean>**

22. **<beans** profile="prod"**>**
23. **<jee:jndi-lookup** id="dataSource" jndi-name="${jpreTest.datasource}"
24. resource-ref="true" **/>**
25. **<jee:jndi-lookup** id="dataSource2" jndi-name="${jpreTest.datasource2}"
26. resource-ref="true" **/>**
27. **</beans>**
29. **<beans** profile="test"**>**
31. **<bean** id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"**>**
32. **<property** name="driverClass" value="${jpreTest.connection.driver\_class}" **/>**
33. **<property** name="jdbcUrl" value="${jpreTest.connection.url}" **/>**
34. **<property** name="user" value="${jpreTest.connection.username}" **/>**
35. **<property** name="password" value="${jpreTest.connection.password}" **/>**
36. <!-- 初始线程数 -->
37. **<property** name="initialPoolSize" value="5" **/>**
38. <!-- 最小线程数 -->
39. **<property** name="minPoolSize" value="${jpreTest.minPoolSize}" **/>**
40. <!-- 最大线程数 -->
41. **<property** name="maxPoolSize" value="${jpreTest.maxPoolSize}" **/>**
42. <!-- 空闲时间，单位秒 -->
43. **<property** name="maxIdleTime" value="1200" **/>**
44. <!-- 获取连接失败后该数据源将申明已断开并永久关闭 -->
45. **<property** name="breakAfterAcquireFailure" value="true" **/>**
46. <!--每60秒检查所有连接池中的空闲连接。Default: 0 -->
47. **<property** name="idleConnectionTestPeriod" value="60"**/>**
48. <!--如果设为true那么在取得连接的同时将校验连接的有效性。Default: false -->
49. **<property** name="testConnectionOnCheckin" value="true" **/>**
50. <!-- 链接池耗尽时一次获取的连接数 -->
51. **<property** name="acquireIncrement" value="5" **/>**
52. **</bean>**
54. **<bean** id="dataSource2" class="com.mchange.v2.c3p0.ComboPooledDataSource"**>**
55. **<property** name="driverClass" value="${jpreTest.connection.driver\_class}" **/>**
56. **<property** name="jdbcUrl" value="${jpreTest.connection.url2}" **/>**
57. **<property** name="user" value="${jpreTest.connection.username2}" **/>**
58. **<property** name="password" value="${jpreTest.connection.password2}" **/>**
59. <!-- 初始线程数 -->
60. **<property** name="initialPoolSize" value="5" **/>**
61. <!-- 最小线程数 -->
62. **<property** name="minPoolSize" value="${jpreTest.minPoolSize}" **/>**
63. <!-- 最大线程数 -->
64. **<property** name="maxPoolSize" value="${jpreTest.maxPoolSize}" **/>**
65. <!-- 空闲时间，单位秒 -->
66. **<property** name="maxIdleTime" value="1200" **/>**
67. <!-- 获取连接失败后该数据源将申明已断开并永久关闭 -->
68. **<property** name="breakAfterAcquireFailure" value="true" **/>**
69. <!--每60秒检查所有连接池中的空闲连接。Default: 0 -->
70. **<property** name="idleConnectionTestPeriod" value="60"**/>**
71. <!--如果设为true那么在取得连接的同时将校验连接的有效性。Default: false -->
72. **<property** name="testConnectionOnCheckin" value="true" **/>**
73. <!-- 链接池耗尽时一次获取的连接数 -->
74. **<property** name="acquireIncrement" value="5" **/>**
75. **</bean>**
76. **</beans>**
77. **</beans>**
78. **</span>**