[spring配置文件详解--真的蛮详细](http://blog.csdn.net/u012049463/article/details/11807587)

转自： <http://book.51cto.com/art/201004/193743.htm>

[此处详细的为我们讲解了spring2.5的实现原理](http://book.51cto.com/art/201004/193508.htm)，感觉非常有用

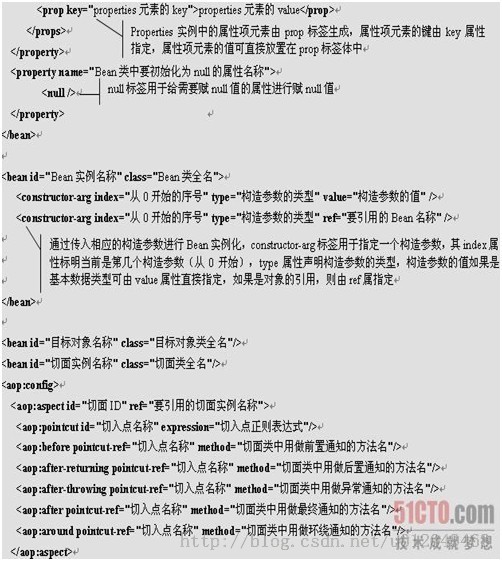
             Spring配置文件是用于指导Spring工厂进行Bean生产、依赖关系注入（装配）及Bean实例分发的"图纸"。Java EE程序员必须学会并灵活应用这份"图纸"准确地表达自己的"生产意图"。Spring配置文件是一个或多个标准的XML文档，applicationContext.xml是Spring的默认配置文件，当容器启动时找不到指定的配置文档时，将会尝试加载这个默认的配置文件。

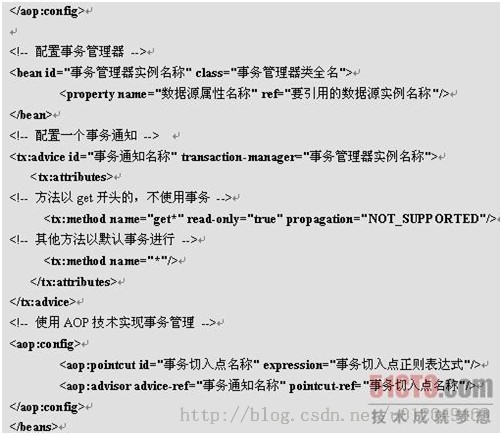
             下面列举的是一份比较完整的配置文件模板，文档中各XML标签节点的基本用途也给出了详细的解释，这些XML标签节点在后续的知识点中均会用到，熟练掌握了这些XML节点及属性的用途后，为我们动手编写配置文件打下坚实的基础。











下面，就上面的配置列举一个示例：

**[html]** [view plaincopy](http://blog.csdn.net/u012049463/article/details/11807587)

1. **<?xml** version="1.0" encoding="UTF-8"**?>**
2. **<beans** xmlns="http://www.springframework.org/schema/beans"
3. xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4. xsi:schemaLocation="http://www.springframework.org/schema/beans
5. http://www.springframework.org/schema/beans/spring-beans-2.5.xsd"**>**
6. <!-- 定义使用C3P0连接池的数据源 -->
7. **<bean** id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"**>**
8. <!-- 指定连接数据库的JDBC驱动 -->
9. **<property** name="driverClass"**>**
10. **<value>**com.mysql.jdbc.Driver**</value>**
11. **</property>**
12. <!-- 连接数据库所用的URL -->
13. **<property** name="jdbcUrl"**>**
14. **<value>**jdbc:mysql://localhost:3306/eportal?useUnicode=
15. true&characterEncoding=gbk**</value>**
16. **</property>**
17. <!-- 连接数据库的用户名 -->
18. **<property** name="user"**>**
19. **<value>**root**</value>**
20. **</property>**
21. <!-- 连接数据库的密码 -->
22. **<property** name="password"**>**
23. **<value>**root**</value>**
24. **</property>**
25. <!-- 设置数据库连接池的最大连接数 -->
26. **<property** name="maxPoolSize"**>**
27. **<value>**20**</value>**
28. **</property>**
29. <!-- 设置数据库连接池的最小连接数 -->
30. **<property** name="minPoolSize"**>**
31. **<value>**2**</value>**
32. **</property>**
33. <!-- 设置数据库连接池的初始化连接数 -->
34. **<property** name="initialPoolSize"**>**
35. **<value>**2**</value>**
36. **</property>**
37. <!-- 设置数据库连接池的连接的最大空闲时间，单位为秒 -->
38. **<property** name="maxIdleTime"**>**
39. **<value>**20**</value>**
40. **</property>**
41. **</bean>**
42. <!-- 定义Hibernate的SessionFactory -->
43. **<bean** id="sessionFactory"
44. class="org.springframework.orm.
45. hibernate3.LocalSessionFactoryBean"**>**
46. <!-- 依赖注入上面定义的数据源dataSource -->
47. **<property** name="dataSource" ref="dataSource" **/>**
48. <!-- 注册Hibernate的ORM映射文件 -->
49. **<property** name="mappingResources"**>**
50. **<list>**
51. **<value>**com/eportal/ORM/News.hbm.xml**</value>**
52. **<value>**com/eportal/ORM/Category.hbm.xml**</value>**
53. **<value>**com/eportal/ORM/Memberlevel.hbm.xml**</value>**
54. **<value>**com/eportal/ORM/Cart.hbm.xml**</value>**
55. **<value>**com/eportal/ORM/Traffic.hbm.xml**</value>**
56. **<value>**com/eportal/ORM/Newsrule.hbm.xml**</value>**
57. **<value>**com/eportal/ORM/Merchandise.hbm.xml**</value>**
58. **<value>**com/eportal/ORM/Admin.hbm.xml**</value>**
59. **<value>**com/eportal/ORM/Orders.hbm.xml**</value>**
60. **<value>**com/eportal/ORM/Cartselectedmer.hbm.xml**</value>**
61. **<value>**com/eportal/ORM/Newscolumns.hbm.xml**</value>**
62. **<value>**com/eportal/ORM/Member.hbm.xml**</value>**
63. **</list>**
64. **</property>**
65. <!-- 设置Hibernate的相关属性 -->
66. **<property** name="hibernateProperties"**>**
67. **<props>**
68. <!-- 设置Hibernate的数据库方言 -->
69. **<prop** key="hibernate.dialect"**>**org.hibernate.dialect.MySQLDialect**</prop>**
70. <!-- 设置Hibernate是否在控制台输出SQL语句，开发调试阶段通常设为true -->
71. **<prop** key="show\_sql"**>**true**</prop>**
72. <!-- 设置Hibernate一个提交批次中的最大SQL语句数 -->
73. **<prop** key="hibernate.jdbc.batch\_size"**>**50**</prop>**
74. **<prop** key="show\_sql"**>**50**</prop>**
75. **</props>**
76. **</property>**
77. **</bean>**
78. <!--定义Hibernate的事务管理器HibernateTransactionManager -->
79. **<bean** id="transactionManager"
80. class="org.springframework.orm.hibernate3.HibernateTransactionManager"**>**
81. <!-- 依赖注入上面定义的sessionFactory -->
82. **<property** name="sessionFactory" ref="sessionFactory" **/>**
83. **</bean>**
84. <!--定义Spring的事务拦截器TransactionInterceptor -->
85. **<bean** id="transactionInterceptor"
86. class="org.springframework.transaction.interceptor.TransactionInterceptor"**>**
87. <!-- 依赖注入上面定义的事务管理器transactionManager -->
88. **<property** name="transactionManager" ref="transactionManager" **/>**
89. <!-- 定义需要进行事务拦截的方法及所采用的事务控制类型 -->
90. **<property** name="transactionAttributes"**>**
91. **<props>**
92. <!-- 以browse、list、load、get及is开头的所有方法采用只读型事务控制类型 -->
93. **<prop** key="browse\*"**>**PROPAGATION\_REQUIRED,readOnly**</prop>**
94. **<prop** key="list\*"**>**PROPAGATION\_REQUIRED,readOnly**</prop>**
95. **<prop** key="load\*"**>**PROPAGATION\_REQUIRED,readOnly**</prop>**
96. **<prop** key="get\*"**>**PROPAGATION\_REQUIRED,readOnly**</prop>**
97. **<prop** key="is\*"**>**PROPAGATION\_REQUIRED,readOnly**</prop>**
98. <!-- 所有方法均进行事务控制，如果当前没有事务，则新建一个事务 -->
99. **<prop** key="\*"**>**PROPAGATION\_REQUIRED**</prop>**
100. **</props>**
101. **</property>**
102. **</bean>**
103. <!-- 定义BeanNameAutoProxyCreatorf进行Spring的事务处理 -->
104. **<bean**
105. class="org.springframework.aop.framework.autoproxy.
106. BeanNameAutoProxyCreator"**>**
107. <!-- 针对指定的bean自动生成业务代理 -->
108. **<property** name="beanNames"**>**
109. **<list>**
110. **<value>**adminService**</value>**
111. **<value>**columnsService**</value>**
112. **<value>**newsService**</value>**
113. **<value>**crawlService**</value>**
114. **<value>**memberLevelService**</value>**
115. **<value>**memberService**</value>**
116. **<value>**categoryService**</value>**
117. **<value>**merService**</value>**
118. **<value>**cartService**</value>**
119. **<value>**ordersService**</value>**
120. **<value>**trafficService**</value>**
121. **</list>**
122. **</property>**
123. <!-- 这个属性为true时，表示被代理的是目标类本身而不是目标类的接口 -->
124. **<property** name="proxyTargetClass"**>**
125. **<value>**true**</value>**
126. **</property>**
127. <!-- 依赖注入上面定义的事务拦截器transactionInterceptor -->
128. **<property** name="interceptorNames"**>**
129. **<list>**
130. **<value>**transactionInterceptor**</value>**
131. **</list>**
132. **</property>**
133. **</bean>**
134. <!-- 装配通用数据库访问类BaseDAOImpl -->
135. **<bean** id="dao" class="com.eportal.DAO.BaseDAOImpl"**>**
136. **<property** name="sessionFactory" ref="sessionFactory" **/>**
137. **</bean>**
138. <!-- 部署系统用户管理业务逻辑组件AdminServiceImpl -->
139. **<bean** id="adminService" class="com.eportal.service.AdminServiceImpl"**>**
140. **<property** name="dao" ref="dao" **/>**
141. **</bean>**
142. <!-- 部署新闻栏目管理业务逻辑组件ColumnsServiceImpl -->
143. **<bean** id="columnsService" class="com.eportal.service.ColumnsServiceImpl"**>**
144. **<property** name="dao" ref="dao" **/>**
145. **</bean>**
147. <!-- 部署订单管理业务逻辑组件OrderServiceImpl -->
148. **<bean** id="ordersService" class="com.eportal.service.OrderServiceImpl"**>**
149. **<property** name="dao" ref="dao" **/>**
150. **</bean>**
151. <!-- 部署流量统计业务逻辑组件TrafficServiceImpl -->
152. **<bean** id="trafficService" class="com.eportal.service.TrafficServiceImpl"**>**
153. **<property** name="dao" ref="dao" **/>**
154. **</bean>**
155. <!-- 部署Struts 2负责系统用户管理的控制器AdminAction -->
156. **<bean** id="adminAction" class="com.eportal.struts.action.
157. AdminAction"
158. scope="prototype"**>**
159. **<property** name="service" ref="adminService" **/>**
160. **</bean>**
161. <!-- 部署Struts 2负责新闻栏目管理的控制器ColumnsAction -->
162. **<bean** id="columnsAction" class="com.eportal.struts.action.
163. ColumnsAction"
164. scope="prototype"**>**
165. **<property** name="service" ref="columnsService" **/>**
166. **</bean>**
167. <!-- 部署Struts 2负责新闻管理的控制器NewsAction -->
168. **<bean** id="newsAction" class="com.eportal.struts.action.
169. NewsAction"
170. scope="prototype"**>**
171. **<property** name="service" ref="newsService" **/>**
172. **<property** name="columnsService" ref="columnsService" **/>**
173. **</bean>**
174. <!-- 部署Struts 2负责新闻采集规则管理的控制器CrawlAction -->
175. **<bean** id="crawlAction" class="com.eportal.struts.action.
176. CrawlAction"
177. scope="prototype"**>**
178. **<property** name="service" ref="crawlService" **/>**
179. **<property** name="columnsService" ref="columnsService" **/>**
180. **</bean>**
182. **</beans>**