这段时间在使用spring+mybatis整合时遇到了一个问题，我想引用外部jdbc.properties，然后在配置文件中使用占位符，如下：

**[html]** [view plain](https://blog.csdn.net/qq_17495033/article/details/48769967) [copy](https://blog.csdn.net/qq_17495033/article/details/48769967)

1. <!--引入属性文件 -->
2. **<context:property-placeholder** location="classpath:jdbc.properties"**/>**
4. <!--数据库连接-->
5. **<bean** id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource"**>**
6. **<property** name="driverClass"**><value>**${jdbc.driverClassName}**</value></property>**
7. **<property** name="jdbcUrl"**><value>**${jdbc.url}**</value></property>**
8. **<property** name="user"**><value>**${jdbc.username}**</value></property>**
9. **<property** name="password"**><value>**${jdbc.password}**</value></property>**
10. **</bean>**

但是我发现出现了以下异常：

**[html]** [view plain](https://blog.csdn.net/qq_17495033/article/details/48769967) [copy](https://blog.csdn.net/qq_17495033/article/details/48769967)

1. WARN [com.mchange.v2.async.ThreadPoolAsynchronousRunner$PoolThread-#2] - Could not load driverClass ${jdbc.driverClassName}
2. ava.lang.ClassNotFoundException: ${jdbc.driverClassName}
3. at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1387)
4. at org.apache.catalina.loader.WebappClassLoader.loadClass(WebappClassLoader.java:1233)
5. at java.lang.Class.forName0(Native Method)
6. at java.lang.Class.forName(Class.java:190)
7. at com.mchange.v2.c3p0.DriverManagerDataSource.ensureDriverLoaded(DriverManagerDataSource.java:100)
8. at com.mchange.v2.c3p0.DriverManagerDataSource.getConnection(DriverManagerDataSource.java:132)
9. at com.mchange.v2.c3p0.WrapperConnectionPoolDataSource.getPooledConnection(WrapperConnectionPoolDataSource.java:182)
10. at com.mchange.v2.c3p0.WrapperConnectionPoolDataSource.getPooledConnection(WrapperConnectionPoolDataSource.java:171)
11. at com.mchange.v2.c3p0.impl.C3P0PooledConnectionPool$1PooledConnectionResourcePoolManager.acquireResource(C3P0PooledConnectionPool.java:137)
12. at com.mchange.v2.resourcepool.BasicResourcePool.doAcquire(BasicResourcePool.java:1014)
13. at com.mchange.v2.resourcepool.BasicResourcePool.access$800(BasicResourcePool.java:32)
14. at com.mchange.v2.resourcepool.BasicResourcePool$AcquireTask.run(BasicResourcePool.java:1810)
15. at com.mchange.v2.async.ThreadPoolAsynchronousRunner$PoolThread.run(ThreadPoolAsynchronousRunner.java:547)

这个问题是无法识别占位符，就是在加载过程中直接把$(jdbc.driverClassName)当做字符串处理了。后来在网上找过问题的解决方式，是说因为配置了

**[html]** [view plain](https://blog.csdn.net/qq_17495033/article/details/48769967) [copy](https://blog.csdn.net/qq_17495033/article/details/48769967)

1. <!-- 扫描接口所在的包  -->
2. **<bean** class="org.mybatis.spring.mapper.MapperScannerConfigurer"**>**
3. **<property** name="basePackage" value="com.xx.dao"**/>**
4. **<property** name="sqlSessionFactory" ref="sqlSessionFactory"**></property>**
5. **</bean>**

MapperScannerConfigurer 先于properties文件处理，意思是没进行占位符的变量赋值就被使用了。去掉sqlSessionFactory这个属性

**[html]** [view plain](https://blog.csdn.net/qq_17495033/article/details/48769967) [copy](https://blog.csdn.net/qq_17495033/article/details/48769967)

1. <!-- 扫描接口所在的包  -->
2. **<bean** class="org.mybatis.spring.mapper.MapperScannerConfigurer"**>**
3. **<property** name="basePackage" value="com.xx.dao"**/>**
4. **</bean>**

同时记得加上这样的bean，id一定是这样，因为要使用的时候会去找这个bean

**[html]** [view plain](https://blog.csdn.net/qq_17495033/article/details/48769967) [copy](https://blog.csdn.net/qq_17495033/article/details/48769967)

1. **<bean** id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean"**>**
2. **<property** name="dataSource" ref="dataSource"**/>**
3. **<property** name="configLocation" value="classpath:config.xml" **/>**
4. **<property** name="mapperLocations"**>**
5. **<value>**classpath:dao/mapper/\*Mapper.xml**</value>**
6. **</property>**
7. **</bean>**

ps:同时注意beans 可能会配置有这个 default-autowire="byName" ，去掉即可。