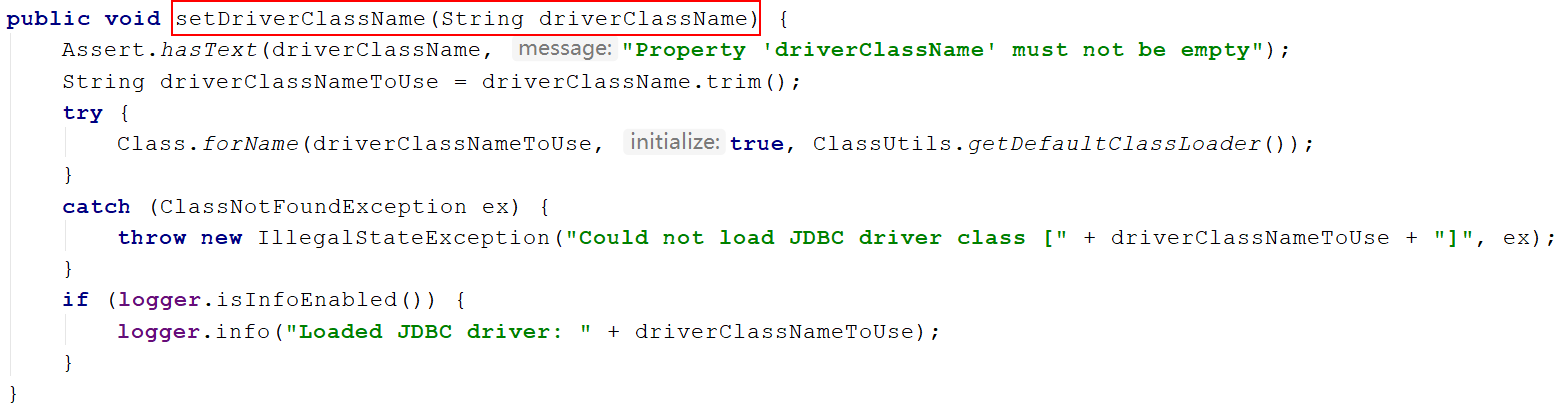
## 1、Spring默认的数据源DriverManagerDataSource

Spring本身提供的一个数据源实现类DriverManagerDataSource，位于org.springframework.jdbc.datasource包中。这个类实现了javax.sql.DataSource接口，但它没有提供池化连接的机制，每次调用getConnection()获取新连接时，只是简单创建一个新的连接。（适用于在单元测试或简单的独立应用中使用，因为它不需要额外的依赖类。）



### driverClassName：用于接收DB驱动。



### url、username、password：数据连接的要素

## 2、DBCP数据源BasicDataSource

Apache下的项目，需要两个jar包。

com.springsource.org.apache.commons.dbcp-1.2.2.osgi.jar

com.springsource.org.apache.commons.pool-1.5.3.jar

DBCP数据源实现类BasicDataSource，其中有driverClassName、url、username、password四个DB连接属性。

一些常用属性：

最大连接数量maxActive=15

最小空闲连接minldle=5

最大空闲连接maxldle=10

初始化连接数initialSize=5

连接被泄漏时是否打印logAbandoned=true

是否自动回收超时连接removeAbandoned=true

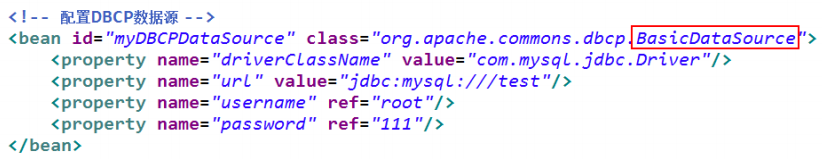
超时时间（以秒为单位）removeAbandonedTimeout=180

最大等待时间maxWait=3000

用来验证从连接池取出的连接validationQuery=SELECT 1 FROM DUAL

用于验证连接是否成功的查询SQL语句，SQL语句必须至少要返回一行数据

指明是否从池中取出连接前进行检查testOnBorrow=true



## 3、C3P0数据源ComboPoolDataSource

C3P0数据源实现类ComboPoolDataSource。

driverClass、jdbcUrl、user、password四个连接属性。



## 4、JNDI数据源

如果应用配置在高性能的服务器（如Weblogic或Websphere等）上，我们可能更希望应用服务器本身提供的数据源。应用服务器的数据源使用JNDI开放调用者使用，Spring为此专门提供引用JNDI资源的JndiObjectFactoryBean类。

<bean id="TerasolunaSampleDataSource"

class="org.springframework.jndi.JndiObjectFactoryBean">

<property name="jndiName">

<value>java:comp/env/jdbc/TerasolunaSampleDataSource</value>

<!-- <value>jdbc/TerasolunaSampleDataSource</value> -->

</property>

</bean>

通过jndiName指定引用的JNDI数据源名称。

Spring2.0为获取J2EE资源提供了一个jee命名空间，通过jee命名空间，可以有效地简化J2EE资源的引用。下面是使用jee命名空间引用JNDI数据源的配置。

<jee:jndi-lookup id="dataSource"

jndi-name="java:comp/env/jdbc/TerasolunaSampleDataSource"/>

</beans>

附在context.xml的DataSource链接情报：

<Context>

<Resource

name="jdbc/TerasolunaSampleDataSource"

type="javax.sql.DataSource"

driverClassName="org.postgresql.Driver"

username="togodb"

password="togo"

url="jdbc:postgresql://127.0.0.1:5432/method\_db"

maxIdle="2"

maxWait="5000"

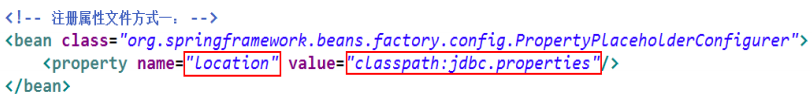
maxActive="4"/>

</Context>

## 5、从属性文件读取数据库连接信息

属性配置在.properties文件中，若该文件要被Spring配置读取，其必须在配置文件中进行注册。两种方式：

### 1）<bean/>方式



### 2）<context:property-placeholder/>方式

