



链滴

spring 整合 mybatis 完整配置

作者: [zml2015](#)

原文链接: <https://ld246.com/article/1500259084634>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

spring-dao.xml

主要配置数据源，采用c3p0连接池

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:tx="http://www.springf
amework.org/schema/tx"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springf
amework.org/schema/beans/spring-beans-2.5.xsd http://www.springframework.org/schema/
x http://www.springframework.org/schema/tx/spring-tx.xsd" >

  <!-- 数据源配置 -->
  <bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource">
    <property name="driverClass">
      <value>${jdbc.driver}</value>
    </property>
    <property name="jdbcUrl">
      <value>${jdbc.url}</value>
    </property>
    <property name="user">
      <value>${jdbc.username}</value>
    </property>
    <property name="password">
      <value>${jdbc.password}</value>
    </property>
    <!--连接池中保留的最小连接数 -->
    <property name="minPoolSize">
      <value>${db.minPoolSize}</value>
    </property>
    <!--连接池中保留的最大连接数。Default: 15 -->
    <property name="maxPoolSize">
      <value>${db.maxPoolSize}</value>
    </property>
    <!--初始化时获取的连接数，取值应在minPoolSize与maxPoolSize之间。Default: 3 -->
    <property name="initialPoolSize">
      <value>${db.initialPoolSize}</value>
    </property>
    <!--连接空闲时间,25000秒内未使用则连接被丢弃，比8小时略微小一些。若为0则永不丢弃， De
ault: 0 -->
    <property name="maxIdleTime">
      <value>${db.maxIdleTime}</value>
    </property>
    <!--当连接池中的连接耗尽的时候c3p0连接同时获取的连接数。Default: 3 -->
    <property name="acquireIncrement">
      <value>${db.acquireIncrement}</value>
    </property>
    <!-- JDBC的标准参数，用以控制数据源内加载的PreparedStatements数量。但由于预缓存的st
tements 属于单个connection而不是整个连接池。所以设置这个参数需要考虑到多方面的因素。
    如果maxStatements与maxStatementsPerConnection均为0，则缓存被关闭。Default: 0 -
  >
    <property name="maxStatements">
      <value>0</value>
```

```

</property>
<!--600秒检查所有连接池中的空闲连接。Default: 0 -->
<property name="idleConnectionTestPeriod">
  <value>600</value>
</property>
<!-- 定义所有连接测试都执行的测试语句。在使用连接测试的情况下这个一显著提高测试速度。
在配置了AutomaticTestTable属性后，此配置失效，Default:
  null -->
<property name="preferredTestQuery">
  <value>select 1 from dual</value>
</property>
<!--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 -->
<property name="testConnectionOnCheckin">
  <value>true</value>
</property>

<!--定义在从数据库获取新连接失败后重复尝试的次数。Default: 30 -->
<property name="acquireRetryAttempts">
  <value>30</value>
</property>

<!-- 获取连接失败将会引起所有等待连接池来获取连接的线程抛出异常。但是数据源仍有效 保
，并在下次调用getConnection()的时候继续尝试获取连接。如果设为true，那么在尝试
  获取连接失败后该数据源将申明已断开并永久关闭。Default: false -->
<property name="breakAfterAcquireFailure">
  <value>false</value>
</property>
</bean>

<!--配置sql session-->
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
  <property name="dataSource" ref="dataSource" />
  <property name="configLocation" value="classpath:mybatis-config.xml"> </property>
  <property name="mapperLocations" value="classpath:com/dtsz/business/**/*.dao/*.xml"
</property>
</bean>
<!-- 配置事务管理器 数据源-->
<bean id="transactionManager"
  class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
  <property name="dataSource" ref="dataSource" />
</bean>
<!--开启事物 只会查找和它在相同的应用上下文件中定义的bean上面的@Transactional注解 -->
<tx:annotation-driven transaction-manager="transactionManager"/>
<!-- 配置 Mapper 所在的包和子包 -->
<bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
  <property name="basePackage" value="top.wuyongshi.business.**.dao" />
  <property name="sqlSessionFactoryBeanName" value="sqlSessionFactory"/>
</bean>

</beans>

```

mybatis-config.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```

<!DOCTYPE configuration
  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>

  <settings>
    <!-- 配置该属性值为 false, 才会保证懒加载好用. -->
    <setting name="aggressiveLazyLoading" value="false"/>

    <!-- 可以将数据库表下划线字段与驼峰类型之间进行转换 -->
    <setting name="mapUnderscoreToCamelCase" value="true"/>

  </settings>

  <plugins>
    <!-- com.github.pagehelper为PageHelper类所在包名 -->
    <plugin interceptor="com.github.pagehelper.PageHelper">
      <property name="dialect" value="mysql"/>
      <!-- 该参数默认为false -->
      <!-- 设置为true时, 会将RowBounds第一个参数offset当成pageNum页码使用 -->
      <!-- 和startPage中的pageNum效果一样-->
      <property name="offsetAsPageNum" value="true"/>
      <!-- 该参数默认为false -->
      <!-- 设置为true时, 使用RowBounds分页会进行count查询 -->
      <property name="rowBoundsWithCount" value="true"/>
      <!-- 设置为true时, 如果pageSize=0或者RowBounds.limit = 0就会查询出全部的结果 -->
      <!-- (相当于没有执行分页查询, 但是返回结果仍然是Page类型) -->
      <property name="pageSizeZero" value="true"/>
      <!-- 3.3.0版本可用 - 分页参数合理化, 默认false禁用 -->
      <!-- 启用合理化时, 如果pageNum<1会查询第一页, 如果pageNum>pages会查询最后一
-->
      <!-- 禁用合理化时, 如果pageNum<1或pageNum>pages会返回空数据 -->
      <!-- <property name="reasonable" value="true"/>-->
      <!-- 3.5.0版本可用 - 为了支持startPage(Object params)方法 -->
      <!-- 增加了一个`params`参数来配置参数映射, 用于从Map或ServletRequest中取值 -->
      <!-- 可以配置pageNum,pageSize,count,pageSizeZero,reasonable,不配置映射的用默认值
-->
      <property name="params" value="pageNum=start;pageSize=limit;pageSizeZero=zero
reasonable=heli;count=contsql"/>
    </plugin>
  </plugins>

</configuration>

```

配置注解扫描

不仅仅是service, 而是包含所有的扫描,还可以将扫描包下某些特定类剔除掉, 如下面的exclude-filter

```

<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:context="http://www.springframework.org/schema/context"
  xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springf

```

```
amework.org/schema/beans/spring-beans.xsd http://www.springframework.org/schema/con
ext http://www.springframework.org/schema/context/spring-context.xsd" >
```

```
<!-- 整合 SpringMVC 1. 配置当前的 SpringIOC 容器的扫描的包 2. 配置 SpringMVC IOC 容器
描的包. -->
<!--非@Controller,扫描@Service @Component @Repository 非仅Service层-->
<context:component-scan base-package="top.wuyongshi.business">
    <context:exclude-filter type="annotation" expression="org.springframework.stereotype.
ontroller" />
    <context:exclude-filter type="annotation" expression="org.springframework.web.bind.
nnotation.ControllerAdvice" />
</context:component-scan>
<context:component-scan base-package="top.wuyongshi.task"/>
</beans>
```

spring.xml

主要用于整合各个配置文件

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:task="http://www.spr
igframework.org/schema/task"
    xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springf
amework.org/schema/beans/spring-beans-2.5.xsd http://www.springframework.org/schema/
ask http://www.springframework.org/schema/task/spring-task.xsd"
    default-autowire="byType" default-lazy-init="false">

    <!-- 部署文件定义,引入配置文件 -->
    <bean id="propertyConfigurer"
        class="org.springframework.beans.factory.config.PreferencesPlaceholderConfigurer">
        <property name="ignoreResourceNotFound" value="true" />
        <property name="locations">
            <list>
                <value>classpath:config.properties</value>
                <value>classpath:db.properties</value>
            </list>
        </property>
    </bean>

    <task:annotation-driven/>

    <!-- 导入其他配置文件 -->
    <import resource="spring-mongodb.xml"/>
    <import resource="spring-dao.xml" />
    <import resource="spring-service.xml" />
    <import resource="spring-task.xml"/>
    <import resource="spring-aop.xml"/>

</beans>
```

db.properties

简单给一下数据库的配置文件

```
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://115.28.167.152:3306/test?useUnicode=true&characterEncoding=UTF-
&zeroDateTimeBehavior=convertToNull
jdbc.username=root
jdbc.password=123456
db.minPoolSize=3
db.maxPoolSize=17
db.initialPoolSize=3
db.maxIdleTime=25000
db.acquireIncrement=2
```