



链滴























1、mybatis 与 spring 整合























作者: [xuguolong](#)

原文链接: <https://ld246.com/article/1542617380805>

来源网站: 链滴

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

一、mybatis 与 spring 的基本整合

1、导入 mybatis 与 spring 整合 jar 包

mybatis 与 spring 的整合包 mybatis-spring 是由 mybatis 团队以及 mybatis 社区提供的

```
org.mybatisgroupId>  
mybatis-springartifactId>  
x.x.xversion>  
dependency>
```

2、配置 SqlSessionFactoryBean

在 mybatis 的使用中，我们通过 SqlSessionFactoryBuilder 来构建一个 SqlSessionFactory 对，而在 spring 整合包中，我们需要使用 SqlSessionFactoryBean 来创建 SqlSessionFactory 的 spring bean 实例。

bean>

dataSource: 这个 datasource 是 spring 容器管理的数据源

3、配置 Mapper

在 mybatis 中，我们的 mapper 通常都是在 mybatis-config.xml 配置文件中，通过节点类指定 mapper 配置文件或者类的位置。在 mybatis-spring 中，通过 MapperFactoryBean 来创建 Mapper 对象

mapperInterface: 我们自定义 mapper 接口的类全路径

sqlSessionFactory: mybatis 的 SqlSessionFactory 对象实例，其实就是第 3 步配置的 sqlSessionFactory，用来创建 SqlSession，然后通过 SqlSession.getMapper(xxx.class) 获取 Mapper 对象

基于扫描方式

上面是单个 mapper 配置方式，但是一个项目中往往是有 mapper 接口的，如果采用这种方式必然会不是很方便，因此 mybatis-spring 提供了基于 package 包扫描的方式来自动扫描和注册这些 mapper

basePackage: 指定需要扫描的包路径，可以同时指定多个 package，使用英文逗号","隔开

```
<blockquote>  
<p>MapperScannerConfigurer 可以指定 SqlSessionFactory 属性，也可以不指定，如果不指定，spring 会自动从容器中查找 SqlSessionFactory 类型的 bean</p>  
</blockquote>  
<p>经过以上几步的配置，mybatis 与 spring 的整合环境就搭建起来了，其他的开发步骤就同普通的 mybatis 项目了。</p>  
<h3 id="二-SqlSessionFactoryBean的配置详解">二、SqlSessionFactoryBean 的配置详解</h3>  
<ul>  
<li>dataSource</li>  
</ul>  
<p>mybatis 与 spring 整合时，mybatis-config.xml 基础配置文件中的会被覆盖，因为需要使用 spring 容器中的 datasource 实例才能进行事务控制</p>  
<ul>  
<li>configLocation</li>  
</ul>  
<p>mybatis-spring 支持 mybatis-config.xml 的配置，通过该属性来指定 mybatis-config.xml 配置文件的位置</p>  
<ul>  
<li>mapperLocations</li>
```


<p>如果是基于 mybatis, 例如: @Select、@Update 等注解开发, 那么可以直接配置 MapperFactoryBean, 但是如果你的 sql 是配置在 xxxMapper.xml 配置文件中, 则需要指定 xxxMapper.xml 配置文件的路径。可以通过 mapperLocations 属性类配置 xxxMapper.xml 配置文件路径, 或者直接在 mybatis-config.xml 文件中指定</p>

<p>其实 mapperLocations 用来简化 mybatis-config.xml 配置, 因为大部分情况下, 我们可能需要 mapper 配置就可以了, 就没必要多创建一个 mybatis-config.xml 配置文件</p>

Configuration

<p>从 mybatis-spring 1.3.0 版本开始, 它也支持直接配置一个 Configuration</p>

其他属性

<p>SqlSessionFactoryBean 支持配置 mybatis-config.xml 中的大部分属性, 例如: typeAliasesPackage、objectFactory 等等, 当我们没有使用 mybatis-config.xml 配置文件时, 可以配置 SqlSessionFactoryBean 的这些属性</p>

<h3 id="三-事务管理">三、事务管理</h3>

<p>经过上面整合后, 就可以直接支持 spring 的事务管理, 我们只需要配置 spring 事务管理器</p>

<h3 id="四-总结">四、总结</h3>

<p>1、配置 sqlSessionFactoryBean 1) 构建 Configuration 2) 构建 sqlSessionFactory 对象</p>

<p>2、配置 MapperBean 1) 不管是基于 mybatis 注解还是 xxxMapper.xml 映射器配置文件, 需要配置 MapperBean 2) 如果是基于 xxxMapper.xml 映射器配置文件开发, 那么还需要通过 mapperLocations 属性或者在 mybatis-config.xml 指定 xxxMapper.xml 配置文件路径</p>

<p>3、DefaultSqlSession 不是线程安全的, 在 mybatis-spring 使用 SqlSessionTemplate 代替 DefaultSqlSession, SqlSessionTemplate 是线程安全的, 所以它产生的 Mapper 也是安全</p>