Mybatis进阶

T A H N K Y O U F O R W A T C H I N G

主讲老师Lison：525765982

课程咨询依娜老师：2470523467

目 录

CONTENTS

与spring的集成

集成配置最佳实践

集成的原理分析

代码生成器 MBG

XML配置详解

怎么运行MGB

关联查询

一对一

一对多

多对多

缓存

一级缓存

二级缓存

MyBatis-Spring是什么

- **Mybatis-spring** 用于帮助你将 MyBatis 代码无缝地整合到 Spring 中。
  - ✓ Spring 将会加载必要的 MyBatis 工厂类和 session 类
  - ✓ 提供一个简单的方式来注入 MyBatis 数据映射器和 SqlSession 到业务层的 bean 中。
  - ✓ 方便集成spring事务
  - ✓ 翻译 MyBatis 的异常到 Spring 的 DataAccessException 异常(数据访问异常)中。
- **Mybatis-spring** 兼容性

MyBatis-Spring要求Java5及以上版本还有下面列出的MyBatis和Spring版本：

MyBatis-Spring	MyBatis	Spring
1.0.0 或 1.0.1	3.0.1 到 3.0.5	3.0.0 或以上
1.0.2	3.0.6	3.0.0 或以上
1.1.0	3.1.0 或以上	3.0.0 或以上

集成配置最佳实践

1. 准备spring项目一个
2. 在pom文件中添加mybatis-spring的依赖

```
<dependency>
        <groupId>org.mybatis</groupId>
        <artifactId>mybatis-spring</artifactId>
        <version>1.3.0</version>
</dependency>
```

3. 配置SqlSessionFactoryBean
4. 配置MapperScannerConfigurer
5. 配置事务

SqlSessionFactoryBean

- 在 MyBatis-Spring 中， SqlSessionFactoryBean 是用于创建 Sql SessionFactory 的。
  - dataSource ：用于配置数据源，该属性为必选项，必须通过这个属性配置数据源 ，这里使用了上一节中配置好的 dataSource 数据库连接池 。
  - mapper Locations ： 配置 SqlSessionFactoryBean 扫描 XML 映射文件的路径，可以使用 Ant 风格的路径进行配置。
  - configLocation ：用于配置mybatis config XML的路径，除了数据源外，对MyBatis的各种配直仍然可以通过这种方式进行，并且配置MyBatis settings 时只能使用这种方式。但配置文件中**任意环境,数据源 和 MyBatis 的事务管理器**都会被忽略；
  - typeAliasesPackage ： 配置包中类的别名，配置后，包中的类在 XML 映射文件中使用时可以省略包名部分 ，直接使用类名。这个配置不支持 Ant风格的路径，当需要配置多个包路径时可以使用分号或逗号进行分隔 。

MapperScannerConfigurer

- 通过 MapperScannerConfigurer类自动扫描所有的 Mapper 接口，使用时可以直接注入接口 。

  MapperScannerConfigurer中常配置以下两个属性 。

  - basePackage ： 用于配置基本的包路径。可以使用分号或逗号作为分隔符设置多于一个的包路径，每个映射器将会在指定的包路径中递归地被搜索到 。
  - annotationClass ： 用于过滤被扫描的接口，如果设置了该属性，那么 MyBatis 的接口只有包含该注解才会被扫描进去

集成的原理分析

1. SqlSessionFactoryBean源码分析
2. MapperScannerConfigurer源码分析

目 录

CONTENTS

与spring的集成

集成配置最佳实践

集成的原理分析

代码生成器 MBG

XML配置详解

怎么运行MGB

关联查询

一对一

一对多

多对多

缓存

一级缓存

二级缓存

Mybatis Generator (MBG)

- **MyBatis Generator：**MyBatis 的开发团队提供了一个很强大的代码生成器，代码包含了数据库表对应的实体类 、Mapper 接口类、 Mapper XML 文件和 Example 对象等，这些代码文件中几乎包含了全部的单表操作方法，使用 MBG 可以极大程度上方便我们使用 MyBatis，还可以减少很多重复操作；
  - generatorConfiguration – 根节点
    - properties – 用于指定一个需要在配置中解析使用的外部属性文件；
    - classPathEntry - 在MBG工作的时候，需要额外加载的依赖包；
    - context -用于指定生成一组对象的环境
      - property (0 个或多个) - 设置一些固定属性
      - plugin (0 个或多个) - 定义一个插件，用于扩展或修改通过 MBG 生成的代码
      - commentGenerator (0 个或 1 个) - 该标签用来配置如何生成注释信息
      - jdbcConnection (1 个) - 必须要有的，使用这个配置链接数据库
      - javaTypeResolver (0 个或 1 个) - 指定 JDBC 类型和 Java 类型如何转换
      - javaModelGenerator (1 个) - java模型创建器
      - sqlMapGenerator (0 个或 1 个) - 生成SQL map的XML文件生成器
      - javaClientGenerator (0 个或 1 个) - 生成Mapper接口
      - table (1个或多个) -选择一个table来生成相关文件，可以有一个或多个table

怎么运行MGB

- ✓ 从命令提示符 使用 XML 配置文件

  java -jar mybatis-generator-core-x.x.x.jar -configfile generatorConfig.xml

- ✓ 作为 Maven Plugin

  mvn mybatis-generator:generate

- ✓ 从另一个 Java 程序 使用 XML 配置文件

目 录

CONTENTS

与spring的集成

集成配置最佳实践

集成的原理分析

代码生成器 MBG

XML配置详解

怎么运行MGB

关联查询

一对一

一对多

多对多

缓存

一级缓存

二级缓存

关联查询

在关系型数据库中，我们经常要处理一对一 、 一对多的关系 。 例如， 一辆汽车需要有一个引擎，这是一对一的关系。 一辆汽车有 4 个或更多个轮子，这是一对多的关系 。关联元素就是专门用来处理关联关系的；

- 关联元素
  - ✓ association　一对一关系
  - ✓ collection　一对多关系
  - ✓ discriminator　鉴别器映射
- 关联方式
  - ✓ 嵌套结果:使用嵌套结果映射来处理重复的联合结果的子集
  - ✓ 嵌套查询:通过执行另外一个 SQL 映射语句来返回预期的复杂类型

一对一 嵌套结果

- association标签 嵌套结果方式 常用属性：
  - ✓ property ：对应实体类中的属性名，必填项。
  - ✓ javaType ： 属性对应的 Java 类型 。
  - ✓ resultMap ： 可以直接使用现有的 resultMap ，而不需要在这里配置映射关系。
  - ✓ columnPrefix ：查询列的前缀，配置前缀后，在子标签配置 result 的 column 时可以省略前缀

Tips:

1. resultMap可以通过使用extends实现继承关系，简化很多配置工作量；
2. 通过添加完整的命名空间，可以引用其他xml文件的resultMap

一对一 嵌套查询

- association标签 嵌套查询方式 常用属性：
  - ✓ select ：另 一个映射查询的 id, MyBatis 会额外执行这个查询获取嵌套对象的结果 。
  - ✓ column ：列名（或别名），将主查询中列的结果作为嵌套查询的 参数，配置方式如 column={propl=coll , prop2=col2}, propl 和 prop2 将作为嵌套查询的参数。
  - ✓ fetchType ：数据加载方式，可选值为 lazy 和 eager，分别为延迟加载和积极加载 ，这个配置会覆盖全局的 lazyLoadingEnabled 配置；

Tips：“N+1 查询问题” 使用“fetchType=lazy”或者全局setting进行改善

概括地讲,N+1 查询问题可以是这样引起的:

- ✓ 你执行了一个单独的 SQL 语句来获取结果列表(就是“+1”)。
- ✓ 对返回的每条记录,你执行了一个查询语句来为每个加载细节(就是“N”)。

这个问题会导致成百上千的 SQL 语句被执行。这通常不是期望的。

一对多

1. collection 支持的属性以及属性的作用和 association 完全相同
2. mybatis会根据id标签，进行字段的合并，合理配置好ID标签可以提高处理的效率；

Tips:

如果要配置一个相当复杂的映射，一定要从基础映射开始配置，每增加一些配置就进行对应的测试，在循序渐进的过程中更容易发现和解决问题 。

discriminator 鉴别器映射

有时一个单独的数据库查询也许返回很多不同 (但是希望有些关联) 数据类型的结果集。 鉴别器元素就是被设计来处理这个情况的, 还有包括类的继承层次结构。 鉴别器非常容易理 解,因为它的表现很像 Java 语言中的 switch 语句；

- discriminator 标签常用的两个属性如下：
  - column：该属性用于设置要进行鉴别比较值的列 。
  - javaType：该属性用于指定列的类型，保证使用相同的 Java 类型来比较值。
- discriminator 标签可以有1个或多个 case 标签， case 标签包含以下三个属性 。
  - value ： 该值为 discriminator 指定 column 用来匹配的值 。
  - resultMap ： 当column的值和value的值匹配时，可以配置使用resultMap指定的映射，resultMap优先级高于 resultType 。
  - resultType ： 当 column 的值和 value 的值匹配时，用于配置使用 resultType指定的映射。

多对多

- ✓ 嵌套结果:使用嵌套结果映射来处理重复的联合结果的子集
- ✓ 嵌套查询:通过执行另外一个 SQL 映射语句来返回预期的复杂类型

目 录

CONTENTS

与spring的集成

集成配置最佳实践

集成的原理分析

代码生成器 MBG

XML配置详解

怎么运行MGB

关联查询

一对一

一对多

多对多

缓存

一级缓存

二级缓存

缓存 一级缓存

MyBatis 包含一个非常强大的查询缓存特性，使用缓存可以使应用更快地获取数据，避免频繁的数据库交互 ；

- 一级缓存 （也叫应用缓存）：
  - 一级缓存默认会启用，想要关闭一级缓存可以在select标签上配置flushCache=“true” ；
  - 一级缓存存在于 SqlSession 的生命周期中，在同一个 SqlSession 中查询时， MyBatis 会把执行的方法和参数通过算法生成缓存的键值，将键值和查询结果存入一个 Map对象中。如果同一个 SqlSession 中执行的方法和参数完全一致，那么通过算法会生成相同的键值，当 Map 缓存对象中已经存在该键值时，则会返回缓存中的对象;
  - 任何的 INSERT 、UPDATE 、 DELETE 操作都会清空一级缓存；

缓存 二级缓存

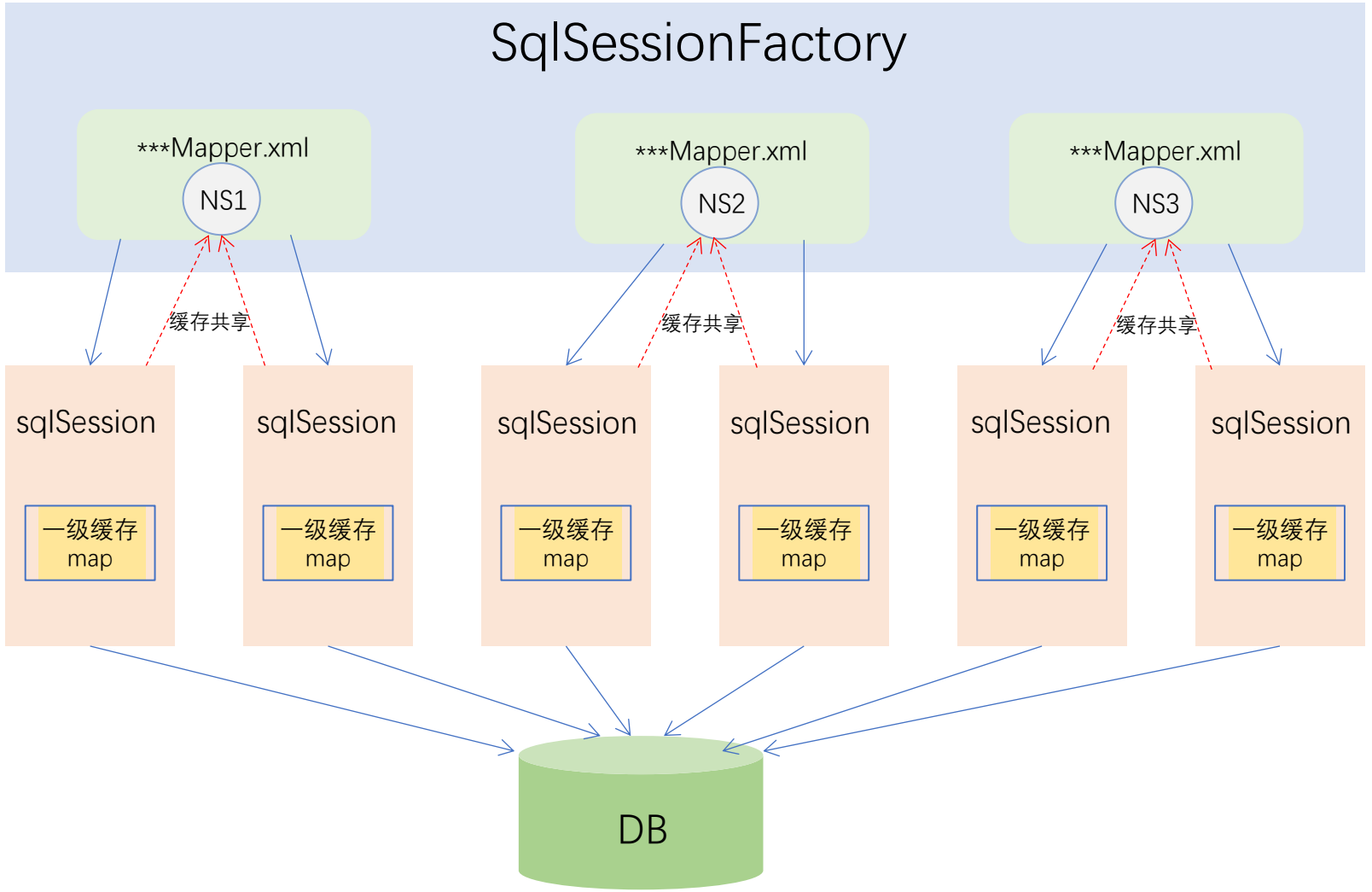
- 二级缓存 （也叫应用缓存）：
  - ✓ 二级缓存存在于 SqlSessionFactory 的生命周期中，可以理解为跨sqlSession；缓存是以namespace为单位的，不同namespace下的操作互不影响。
  - ✓ setting参数 cacheEnabled，这个参数是二级缓存的全局开关，默认值是 true，如果把这个参数设置为 false，即使有后面的二级缓存配置，也不会生效；
  - ✓ 要开启二级缓存,你需要在你的 SQL 映射文件中添加配置：

```
<cache eviction="FIFO" flushInterval="60000" size="512" readOnly="true"/>
```

  - ✓ 字面上看就是这样。这个简单语句的效果如下：
    - 映射语句文件中的所有 select 语句将会被缓存。
    - 映射语句文件中的所有 insert,update 和 delete 语句会刷新缓存。
    - 缓存会使用 Least Recently Used(LRU,最近最少使用的)算法来收回。
    - 根据时间表(比如 no Flush Interval,没有刷新间隔), 缓存不会以任何时间顺序 来刷新。
    - 缓存会存储列表集合或对象(无论查询方法返回什么)的 1024 个引用。
    - 缓存会被视为是 read/write(可读/可写)的缓存,意味着对象检索不是共享的,而 且可以安全地被调用者修改,而不干扰其他调用者或线程所做的潜在修改。

Tips: 使用二级缓存容易出现脏读，建议避免使用二级缓存，在业务层使用可控制的缓存代替更好；

缓存 示意图


SqlSessionFactory
***Mapper.xml
NS1
***Mapper.xml
NS2
***Mapper.xml
NS3
缓存共享
缓存共享
缓存共享
sqlSession
sqlSession
sqlSession
sqlSession
sqlSession
sqlSession
一级缓存 map
一级缓存 map
一级缓存 map
一级缓存 map
一级缓存 map
一级缓存 map
DB