

MyBatis 源码分析----MyBatis 整体架构概要说明

作者：西瓜1994

时间：2018/10/17 09:06

标签：MyBatis

MyBatis整体架构

MyBatis的整体架构分为三层1：基础支持层，2：核心处理层，3：接口层



1:基础支持层：

1-1反射模块：

该模块对Java 原生的反射进行了良好的封装，提供了更加简洁易用的API，方便上层使调用，并且对反射操作进行了一系列优化，例如缓存了类的元数据，提高了反射操作的性能。

1-2类型转换模块：

MyBatis 为简化配置文件提供了别名机制，该机制是类型转换模块的主要功能之一。类型转换模块的另一个功能是实现JDBC 类型与Java 类型之间的转换，该功能在为SQL 语句绑定实参以及映射查询结果集时都会涉及。在为SQL 语句绑定实参时，会将数据由Java 类型转换成JDBC 类型；而在映射结果集时，会将数据由JDBC 类型转换成Java 类型

日志模块：

日志模块的一个主要功能就是集成第三方日志框架，将MyBatis操作的日志输出。

1-3资源加载模块：

资源加载模块主要是对类加载器进行封装，确定类加载器的使用顺序，并提供了加载类文件以及其他资源文件的功能

1-4解析器模块：

解析器模块的主要提供了两个功能：一个功能是对XPath 进行封装，为MyBatis 初始化时解析 mybatis-config.xml 配置文件以及映射配置文件提供支持；另一个功能是为处理动态SQL 语句中的占位符提供支持。

1-5数据源模块：

数据源是实际开发中常用的组件之一。现在开源的数据源都提供了比较丰富的功能，例如，连接池功能、检测连接状态等，选择性能优秀的数据源组件对于提升ORM 框架乃至整个应用的性能都是非常重要的。MyBatis 自身提供了相应的数据源实现，当然MyBatis 也提供了与第三方数据源集成的接口，这些功能都位于数据源模块之中

1-6事务管理：

MyBatis 对数据库中的事务进行了抽象，其自身提供了相应的事务接口和简单实现。

在很多场景中，MyBatis 会与Spring 框架集成，并由Spring 框架管理事务。

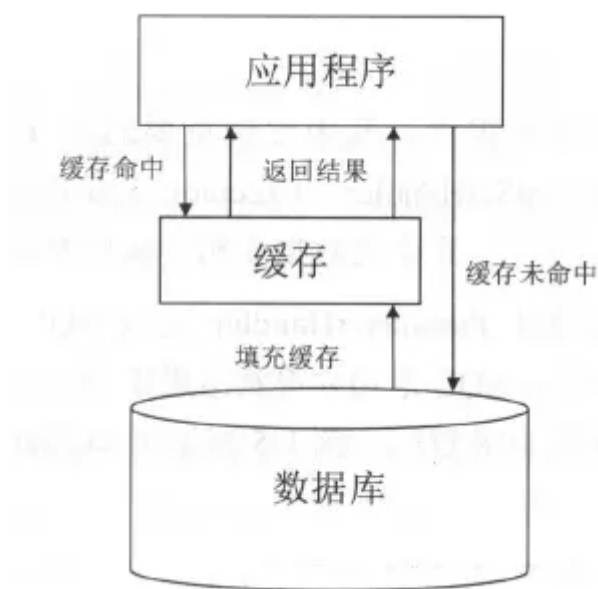
1-7缓存模块：

MyBatis 中提供了一级缓存和二级缓存，而这两级缓存都是依赖于基础支持层中的缓存模块实现的。这里需要读者注意的是，MyBatis 中自带的这两级缓存与MyBatis 以及整个应用是运行在同一个jvm 中的，共享同一块堆内存。如果这两级缓存中的数据量较大，则可能影响系统中其他功能的运行，所以当需要缓存大量数据时，优先考虑使用Redis、Memcache 等缓存产品。

1-8 Binding模块：

在调用Sql Session 相应方法执行数据库操作时，需要指定映射文件中定义的SQL 节点，如果出现拼写错误，我们只能在运行时才能发现相应的异常。为了尽早发现这种错误，MyBatis 通过 Binding 模块将用户自定义的Mapper 接口与映射配置文件关联起来，系统可以通过调用自定义Mapper 接口中的方法执行相应的SQL 语句完成数据库操作，从而避免上述问题。

值得注意的是，开发人员无须编写自定义Mapper 接口的实现，MyBatis 会自动为其创建动态代理对象。在有些场景中，自定义Mapper 接口可以完全代替映射配置文件，但有的映射规则和SQL 语句的定义还是写在映射配置文件中比较方便，例如动态SQL 语句的定义。



2:核心处理层

2-1配置解析：

在MyBatis 初始化过程中，会加载mybatis-config.xml 配置文件、映射配置文件以及Mapper 接口中的注解信息，解析后的配置信息会形成相应的对象并保存到 Configuration 对象中。例如，示例中定义的 < resultMap > 节点（即ResultSet 的映射规则）会被解析成ResultMap 对象：示例中定义的 < result> 节点（即属性映射）会被解析成 ResultMapping 对象。之后，利用该 Configuration 对象创建SqlSessionFactory 对象。待My Batis 初始化之后，开发人员可以通过初始化得到SqlSessionFactory 创建 SqlSession 对象并完成数据库操作。

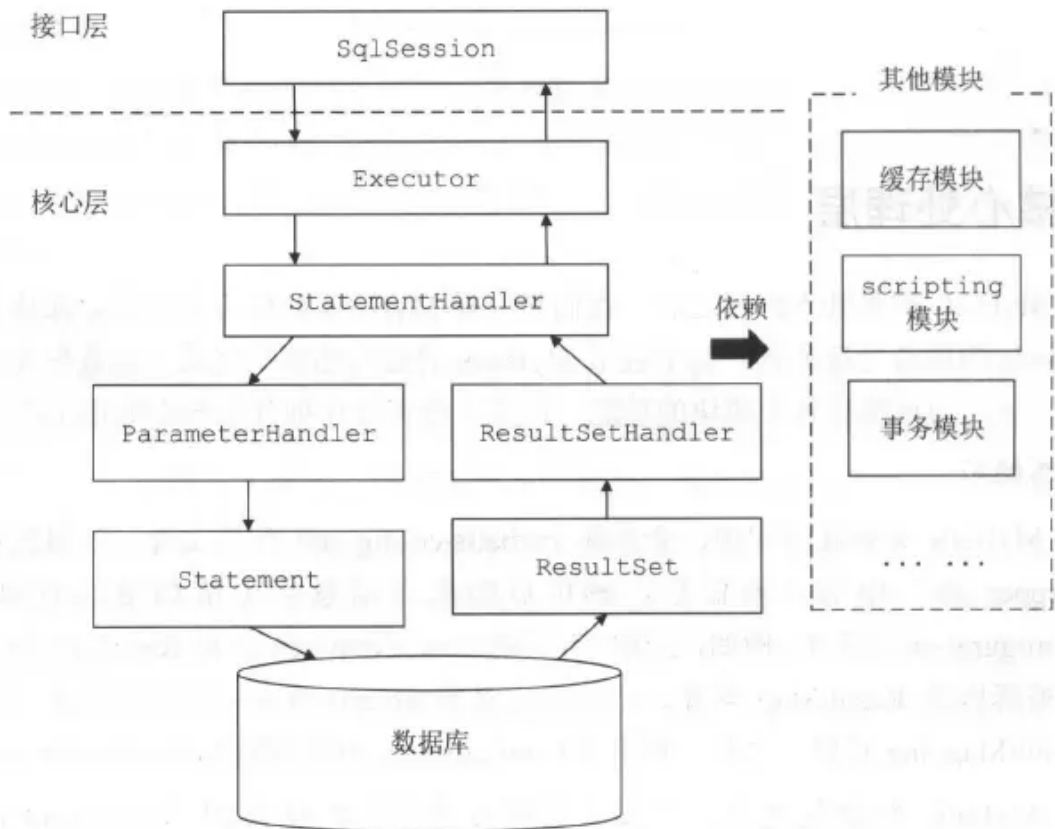
2-2:SQL解析与scripting模块

拼凑SQL 语句是一件烦琐且易出错的过程，为了将开发人员从这项枯燥无趣的工作中解脱出来，MyBatis 实现动态SQL 语句的功能，提供了多种动态SQL 语句对应的节点，例如，< where > 节点、< if > 节点、< foreach > 节点等。通过这些节点的组合使用，开发人员可以写出几乎满足所有需求的动态SQL 语句。

My Batis 中的scripting 模块会根据用户传入的实参，解析映射文件中定义的动态SQL 节点，并形成数据库可执行的SQL 语句。之后会处理SQL 语句中的占位符，绑定用户传入的实参。

2-3:SQL执行

SQL 语句的执行涉及多个组件，其中比较重要的是Executor、StatementHandler、ParameterHandler 和ResultSetHandler。Executor 主要负责维护一级缓存和二级缓存，并提供事务管理的相关操作，它会将数据库相关操作委托给StatementHandler 完成。StatementHandler 首先通过ParameterHandler 完成SQL 语句的实参绑定，然后通过java.sql.Statement 对象执行SQL 语句并得到结果集，最后通过ResultSetHandler 完成结果集的映射，得到结果对象并返回。下图展示了MyBatis 执行一条SQL 语句的大致过程。



2-4：插件

Mybatis 自身的功能虽然强大，但是并不能完美切合所有的应用场景，因此MyBatis 提供了插件接口，我们可以通过添加用户自定义插件的方式对MyBatis 进行扩展。用户自定义插件也可以改变Mybatis 的默认行为，例如，我们可以拦截SQL 语句并对其进行重写。由于用户自定义插件会影响MyBatis 的核心行为，在使用自定义插件之前，开发人员需要了解MyBatis 内部的原理，这样才能编写出安全、高效的插件。

3:接口层

接口层相对简单，其核心是SqlSession 接口，该接口中定义了MyBatis 暴露给应用程序调用的API，也就是上层应用与MyBatis 交互的桥梁。接口层在接收到调用请求时，会调用核心处理层的相应模块来完成具体的数据库操作

本文地址：<https://my.oschina.net/u/3905482/blog/2247731>

© 著作权归作者所有