

Mybatis持久层半自动化框架解析

持久层框架MyBatis

实现解析

分析与实现

JDBC已经可以完成对数据库的操作，为什么还需要持久层框架？

JDBC存在哪些问题和缺陷？

1. JDBC存在硬编码问题：数据库驱动、数据库连接等配置信息无法进行动态修改
2. 频繁创建数据库连接：执行每条sql都需要去创建数据库连接
3. SQL语句、SQL入参、SQL返回结果集都存在硬编码问题
4. 需要手动封装结果集，较为繁琐

持久层框架如何对JDBC的问题进行优化？

1. JDBC的硬编码问题，将数据库驱动信息、数据库连接信息抽取到配置文件中，通过dom4j读取配置文件来获取
2. 使用数据库连接池进行创建数据库连接
3. 将SQL语句、SQL入参、SQL返回结果集都抽取到配置文件中，通过dom4j读取xml配置文件，再通过反射或内串获取sql信息

自定义持久层框架

1. 将数据库连接信息封装至 sqlMapConfig.xml 文件中，同时将 mapper.xml 的路径提取到 sqlMapConfig.xml 中
2. 将 sql 信息封装至 mapper.xml 中，每个 mapper.xml 都有唯一的 namespace，通过 namespace + sqlId 的方式定位到每条 sql 语句

使用端：引入自定义持久层框架的jar包

- 加载配置文件：根据配置文件的路径，加载配置文件成字节输入流，存储在内存中

1. 创建 Resources 类，通过 InputStream 的 getResourceAsStream(String path) 方法将 sqlMapConfig.xml 中的标签内容读取到内存中
2. 创建两个 javaBean，用来存放对配置文件解析出来的内容：Configuration 核心配置类，存放 sqlMapConfig.xml 中解析出来的内容；MappedStatement 映射配置类，存放 mapper.xml 解析出来的内容
3. 解析配置文件，引入 dom4j，创建 SqlSessionFactoryBuilder，通过在类中创建 build(InputStream in) 解析配置文件
- 解析步骤：
 1. 使用 dom4j 解析配置文件，将解析出来的内容封装到容器对象中
 2. 创建 SqlSessionFactory 对象：生产 sqlSession 会话对象（工厂模式）
4. 创建 SqlSessionFactory 接口及实现类 DefaultSqlSessionFactory
5. 创建 SqlSession 接口及实现类 DefaultSqlSession
 - 定义对数据库的CRUD操作
 - selectList()
 - selectOne()
 - update()
 - delete()
6. 创建 Executor 接口及 SimpleExecutor 实现类
 - query(Configuration, MappedStatement, Object... params): 执行的就是JDBC代码

自定义持久层框架 MyBatis

接口层

- 数据增加接口
- 数据删除接口
- 数据查询接口
- 数据修改接口
- 配置信息维护接口

接口调用方式：基于 Statement ID 或 基于Mapper接口

MyBatis 架构原理

数据处理层

- 参数映射
 - 参数映射配置
 - 参数映射解析
 - 参数类型解析

ParameterHandler
- SQL解析
 - SQL语句配置
 - SQL语句解析
 - SQL语句动态生成

SqlSource
- SQL执行
 - SimpleExecutor (默认普通执行器)
 - BatchExecutor (重用语句并批量更新)
 - ReuseExecutor (重用预处理语句 prepare statements)

Executor
- 结果处理和映射
 - 结果映射配置
 - 结果类型解析
 - 结果类型解析

ResultSetHandler

框架支撑层

- 事务管理
- 连接池管理
- 缓存机制

1. 基于XML配置
2. 基于注解

SQL语句配置方式

源码解析

MyBatis 层次结构



MyBatis 执行流程

1. 通过字节流将配置文件加载到内存中，此时尚未解析
2. 解析内存配置文件成 configuration 对象，并创建 DefaultSqlSessionFactory 对象
3. 创建 DefaultSqlSession 对象，并将事务设置为不自动提交，同时实例化 Executor 对象
4. 根据 statementid 在 configuration 的 Map 集合中找到指定的 MappedStatement 对象，并将 sql 执行任务委派给了 Executor 执行器