

一、MyBatis简介

- 1、MyBatis历史
- 2、MyBatis特性
- 3、MyBatis下载
- 4、和其它持久化层技术对比

二、搭建MyBatis

- 1、开发环境
- 2、创建maven工程
 - a>打包方式: jar
 - b>引入依赖
- 3、创建MyBatis的核心配置文件
- 4、创建mapper接口
- 5、创建MyBatis的映射文件
- 6、通过junit测试功能
- 7、加入log4j日志功能
 - a>加入依赖
 - b>加入log4j的配置文件

三、核心配置文件详解

四、MyBatis的增删改查

五、MyBatis获取参数值的两种方式 (重点)

- 1、单个字面量类型的参数
- 2、多个字面量类型的参数
- 3、map集合类型的参数
- 4、实体类类型的参数
- 5、使用@Param标识参数

六、MyBatis的各种查询功能

- 1、查询一个实体类对象
- 2、查询一个list集合
- 3、查询单个数据
- 4、查询一条数据为map集合
- 5、查询多条数据为map集合
 - 方式一:
 - 方式二:

七、特殊SQL的执行

- 1、模糊查询
- 2、批量删除
- 3、动态设置表名
- 4、添加功能获取自增的主键

八、自定义映射resultMap

- 1、resultMap处理字段和属性的映射关系
- 2、多对一映射处理
 - a>级联方式处理映射关系
 - b>使用association处理映射关系
 - c>分步查询
- 3、一对多映射处理
 - a>collection
 - b>分步查询

九、动态SQL

- 1、if
- 2、where
- 3、trim
- 4、choose、when、otherwise
- 5、foreach
- 6、SQL片段

十、MyBatis的缓存

- 1、MyBatis的一级缓存
- 2、MyBatis的二级缓存
- 3、二级缓存的相关配置
- 4、MyBatis缓存查询的顺序
- 5、整合第三方缓存EHCache
 - a>添加依赖
 - b>各jar包功能
 - c>创建EHCache的配置文件ehcache.xml
 - d>设置二级缓存的类型
 - e>加入logback日志
 - f>EHCache配置文件说明

十一、MyBatis的逆向工程

- 1、创建逆向工程的步骤
 - a>添加依赖和插件
 - b>创建MyBatis的核心配置文件
 - c>创建逆向工程的配置文件
 - d>执行MBG插件的generate目标
- 2、QBC查询

十二、分页插件

- 1、分页插件使用步骤
 - a>添加依赖
 - b>配置分页插件
- 2、分页插件的使用

一、MyBatis简介

1、MyBatis历史

MyBatis最初是Apache的一个开源项目*iBatis*，2010年6月这个项目由Apache Software Foundation迁移到了Google Code。随着开发团队转投Google Code旗下，*iBatis3.x*正式更名为MyBatis。代码于2013年11月迁移到Github。

*iBatis*一词来源于“internet”和“abatis”的组合，是一个基于Java的持久层框架。*iBatis*提供的持久层框架包括SQL Maps和Data Access Objects (DAO)。

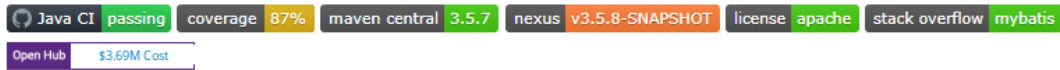
2、MyBatis特性

- 1) MyBatis 是支持定制化 SQL、存储过程以及高级映射的优秀的持久层框架
- 2) MyBatis 避免了几乎所有的JDBC 代码和手动设置参数以及获取结果集
- 3) MyBatis可以使用简单的XML或注解用于配置和原始映射，将接口和Java的POJO (Plain Old Java Objects, 普通的Java对象) 映射成数据库中的记录
- 4) MyBatis 是一个半自动的ORM (Object Relation Mapping) 框架

3、MyBatis下载

MyBatis下载地址: <https://github.com/mybatis/mybatis-3>

MyBatis SQL Mapper Framework for Java



The MyBatis SQL mapper framework makes it easier to use a relational database with object-oriented applications. MyBatis couples objects with stored procedures or SQL statements using a XML descriptor or annotations. Simplicity is the biggest advantage of the MyBatis data mapper over object relational mapping tools.

Essentials

- [See the docs](#)
- [Download Latest](#) 点这里下载最新版
- [Download Snapshot](#)

The screenshot shows the GitHub release page for mybatis-3.5.7. It includes a 'Latest release' badge, the repository name 'mybatis-3.5.7', and the release date '4d57711' by user 'harawata'. A 'Bug fixes' section lists 'Improved performance under JDK 8. #2223'. Below this, it states 'There is no known backward incompatible change since 3.5.6.' and 'Please see the 3.5.7 milestone page for the complete list of changes.' The 'Assets' section contains three items: 'mybatis-3.5.7.zip' (with a link to download all resources), 'Source code (zip)', and 'Source code (tar.gz)'.

4、和其它持久化层技术对比

- JDBC
 - SQL 夹杂在Java代码中耦合度高，导致硬编码内伤
 - 维护不易且实际开发需求中 SQL 有变化，频繁修改的情况多见
 - 代码冗长，开发效率低
- Hibernate 和 JPA
 - 操作简便，开发效率高
 - 程序中的长难复杂 SQL 需要绕过框架
 - 内部自动生产的 SQL，不容易做特殊优化
 - 基于全映射的全自动框架，大量字段的 POJO 进行部分映射时比较困难。
 - 反射操作太多，导致数据库性能下降

- MyBatis
 - 轻量级，性能出色
 - SQL 和 Java 编码分开，功能边界清晰。Java代码专注业务、SQL语句专注数据
 - 开发效率稍逊于Hibernate，但是完全能够接受

二、搭建MyBatis

1、开发环境

IDE: idea 2019.2

构建工具: maven 3.5.4

MySQL版本: MySQL 5.7

MyBatis版本: MyBatis 3.5.7

2、创建maven工程

a>打包方式: jar

b>引入依赖

```
<dependencies>
  <!-- Mybatis核心 -->
  <dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>3.5.7</version>
  </dependency>

  <!-- junit测试 -->
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>test</scope>
  </dependency>

  <!-- MySQL驱动 -->
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.3</version>
  </dependency>
</dependencies>
```

3、创建MyBatis的核心配置文件

习惯上命名为mybatis-config.xml，这个文件名仅仅只是建议，并非强制要求。将来整合Spring之后，这个配置文件可以省略，所以大家操作时可以直接复制、粘贴。

核心配置文件主要用于配置连接数据库的环境以及MyBatis的全局配置信息

核心配置文件存放的位置是src/main/resources目录下

```
<?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>
  <!--设置连接数据库的环境-->
  <environments default="development">
    <environment id="development">
      <transactionManager type="JDBC"/>
      <dataSource type="POOLED">
        <property name="driver" value="com.mysql.jdbc.Driver"/>
        <property name="url"
value="jdbc:mysql://localhost:3306/MyBatis"/>
        <property name="username" value="root"/>
        <property name="password" value="123456"/>
      </dataSource>
    </environment>
  </environments>
  <!--引入映射文件-->
  <mappers>
    <mapper resource="mappers/UserMapper.xml"/>
  </mappers>
</configuration>
```

4、创建mapper接口

MyBatis中的mapper接口相当于以前的dao。但是区别在于，mapper仅仅是接口，我们不需要提供实现类。

```
public interface UserMapper {

  /**
   * 添加用户信息
   */
  int insertUser();

}
```

5、创建MyBatis的映射文件

相关概念：**ORM**（Object Relationship Mapping）对象关系映射。

- 对象：Java的实体类对象
- 关系：关系型数据库
- 映射：二者之间的对应关系

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/546100033013010125>