

版权注意事项：

- 1、书籍版权归作者和出版社所有
- 2、本PDF仅限用于个人获取知识，进行私底下的知识交流
- 3、PDF获得者不得在互联网上以任何目的进行传播
- 4、如觉得书籍内容很赞，请购买正版实体书，支持作者
- 5、请于下载PDF后24小时内删除本PDF。

▶ 中国电子教育学会高教分会推荐
高等学校应用型本科“十三五”规划教材

机电 MECHATRONICS
计算机 COMPUTER
电子 ELECTRONICS

- 突破传统的侧重技术细节的形式，以“项目驱动、任务导向”的方式组织内容。
- 配有案例源代码与课件，方便使用。

Java EE 框架技术

(SpringMVC+Spring+MyBatis)

▶ 主 编 陈永政 张正龙
副主编 谢东亮 张红实 李冀明

中国电子科技集团公司第38研究所

高等学校计算机专业“十三五”规划教材

Java EE 框架技术

(Spring 4.0 / MyBatis 3.2)

站在巨人的肩上

Standing on Shoulders of Giants

王 滔 陈永斌 张正虎

副主编 谢克忠 徐红波 李英明

西安电子科技大学出版社

中国电子教育学会高教分会推荐

高等学校应用型本科“十三五”规划教材

Java EE框架技术

(SpringMVC + Spring + MyBatis)

主 编 陈永政 张正龙

副主编 谢东亮 张红实 李冀明

西安电子科技大学出版社

内 容 简 介

本书对当前企业使用较多的、流行的三大技术框架 SpringMVC、Spring 和 MyBatis 的基本知识和使用方法进行了详细的讲解。全书共分为 7 章。第一章主要介绍 MyBatis 开发入门知识；第二章主要介绍 MyBatis 配置选项；第三章主要介绍 MyBatis 映射器(Mapper)；第四章主要介绍 Spring 核心技术；第五章主要介绍 SpringMVC；第六章主要介绍 SpringMVC、Spring、MyBatis 三个框架的集成；第七章主要是项目实战部分。本书在讲解知识点的同时还提供了丰富的案例，每章节末均给出一定量的练习题，以帮助学生巩固学习效果，加深对相关知识的理解。

本书可作为高等院校计算机相关专业软件工程类课程的教材，也可作为相关工程技术人员的参考用书。

图书在版编目(CIP)数据

Java EE 框架技术：SpringMVC+Spring+MyBatis/陈永政，张正龙主编。

—西安：西安电子科技大学出版社，2017.8

ISBN 978-7-5606-4589-6

I. ① J… II. ① 陈… ② 张… III. ① JAVA 语言—程序设计 IV. ① TP312

中国版本图书馆 CIP 数据核字(2017)第 171756 号

策 划 李惠萍

责任编辑 黄茵 阎彬

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西利达印务有限责任公司

版 次 2017 年 8 月第 1 版 2017 年 8 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 14

字 数 325 千字

印 数 1~3000 册

定 价 26.00 元

ISBN 978-7-5606-4589-6/TP

XDUP 4881001-1

如有印装问题可调换

前 言

本书介绍的 SpringMVC、Spring、MyBatis 开源框架开发技术，是基于 Java EE 的 Web 应用程序，用于开发企业级 Web 应用的软件框架。Java EE 通过提供中间层集成框架来满足各种应用需求。Java EE 架构具有高可用性、高可靠性、高扩展性，并且成本低，是企业构建 Web 应用平台的首选。而 Java EE 架构通常选用 SpringMVC+Spring+MyBatis 框架作为其基础开发框架。通过对三个框架的合理整合，不仅可以大幅度提高系统的开发效率，而且能提高系统的稳定性、健壮性与安全性。

本书介绍了 Java EE 开发使用的三大开发框架 SpringMVC、Spring、MyBatis 及其整合使用。本书内容共分为七章。第一章为 MyBatis 开发入门知识，讨论了 MyBatis 的优势，使用 MyBatis 访问数据库的优点，并创建了第一个 MyBatis 项目，带领读者进入 MyBatis。第二章为 MyBatis 配置选项，使用基于 XML 配置和基于 Java API 配置的方式引导 MyBatis。第三章为映射器配置，是本书重点，讨论了怎样使用映射器配置文件书写 SQL 映射语句，如何配置简单的语句、一对一以及一对多关系的语句，以及怎样使用 resultMap 进行结果集映射；还讨论了动态 SQL 的书写方法及使用注解书写 SQL 映射语句，最后介绍如何使用 MyBatis Generator 自动创建实体类、接口、配置文件代码。第四章为 Spring 核心技术，讨论了 Spring 的核心知识，包括 Spring IoC 和 Spring AOP 技术。第五章为 SpringMVC 框架技术，包括 SpringMVC 概述、创建第一个 SpringMVC 程序、SpringMVC RequestMapping 的基本设置、SpringMVC 前后台数据交互、SpringMVC 文件上传下载，以及 SpringMVC 常用注解。第六章为 SpringMVC、Spring、MyBatis 的集成部分，介绍了三个框架的集成步骤。第七章为项目实战部分，主要以云服务器租赁后台管理系统为导向，介绍了综合应用 SpringMVC+Spring+MyBatis 框架实现一个项目的技术与过程。

本书突破传统的侧重 Java EE 技术细节介绍的形式，以“项目驱动、任务导向”的方式进行内容组织。首先以项目案例的实现为先导，让读者了解某项技术的应用，引起读者对这些技术实现的兴趣，激起其探索该技术实现原理与理论知识的愿望。然后通过有目的的学习，让读者掌握书中介绍的知识点及实现技术。本书介绍的相关技术具有连贯性。

本书适合作为高等院校计算机相关专业软件工程类课程的教材，也适合作为相关工程技术人员的参考用书。本书配有一系列案例源代码，这些案例代码均经过调试可以运行。书中介绍了这些案例的实现过程，读者可以按照书中介绍的案例实现步骤自行实现，并可借助这些案例引导，逐步掌握使用 SpringMVC、Spring、MyBatis 框架进行综合应用软件项目的开发。本书相关源码下载地址：<https://github.com/bay-test/ssmbooksource>。

本书由陈永政和张正龙担任主编，陈永政主要承担了第一章、第二章、第三章、第五章、第六章的编写；张正龙主要承担了第四章、第七章的编写；谢东亮、张红实、李冀明承担了部分章节的编写，并提出了大量有益的建议。重庆知人者科技有限公司的沈国防工程师参与了本书教学案例的设计及教学内容的设计，在此一并表示感谢。

由于时间仓促及编者水平有限，书中难免存在疏漏和不足之处，恳请同行专家和读者给予批评与指正。

编者邮箱：610919606@qq.com

编者

2017年5月

目 录

第一章 MyBatis 开发入门.....	1
1.1 MyBatis 简介.....	1
1.2 MyBatis 的优势.....	2
1.3 认识第一个 MyBatis 程序.....	3
1.4 MyBatis 日志.....	8
1.4.1 MyBatis 日志的实现方式.....	8
1.4.2 使用 Log4J 实现 MyBatis 日志的配置.....	9
本章小结.....	10
练习题.....	11
第二章 配置 MyBatis.....	13
2.1 基于 XML 方式配置 MyBatis.....	13
2.1.1 属性 properties.....	15
2.1.2 全局参数设置 settings.....	16
2.1.3 类型别名 typeAliases.....	18
2.1.4 类型处理器 typeHandlers.....	20
2.1.5 环境集合属性对象 environments.....	25
2.1.6 映射器 mappers.....	28
2.1.7 对象工厂 ObjectFactory.....	28
2.1.8 插件 plugins.....	30
2.2 基于 Java API 方式配置 MyBatis.....	31
2.2.1 环境配置 Environment.....	32
2.2.2 类型别名 typeAliases.....	33
2.2.3 类型处理器 typeHandlers.....	34
2.2.4 全局参数设置 Settings.....	34
2.2.5 映射器 mappers.....	35
本章小结.....	35
练习题.....	36
第三章 映射器(Mapper).....	38
3.1 SQL 映射配置文件和 SQL 映射接口.....	38
3.2 SQL 映射.....	43
3.2.1 select 查询语句.....	44
3.2.2 insert 插入语句.....	46

3.2.3	update 修改语句.....	48
3.2.4	delete 删除语句.....	49
3.2.5	SQL 块语句.....	50
3.2.6	Parameters 参数.....	50
3.2.7	resultMap 结果集映射.....	52
3.3	SQL 高级映射.....	57
3.3.1	拓展 resultMap.....	57
3.3.2	一对一映射.....	57
3.3.3	一对多映射.....	63
3.3.4	cache 和 cache-ref 元素.....	67
3.4	动态 SQL.....	68
3.4.1	if 元素.....	69
3.4.2	choose、when、otherwise 元素.....	69
3.4.3	where、trim、set 元素.....	70
3.4.4	foreach 元素.....	72
3.5	注解配置 SQL 映射器.....	73
3.5.1	@Select 查询语句.....	73
3.5.2	@Insert 插入语句.....	74
3.5.3	@Update 修改语句.....	75
3.5.4	@Delete 删除语句.....	76
3.5.5	@ResultMap 结果映射.....	76
3.5.6	@One 一对一映射.....	78
3.5.7	@Many 一对多映射.....	79
3.5.8	@SelectProvider 动态查询语句.....	80
3.5.9	@InsertProvider 动态插入语句.....	83
3.5.10	@UpdateProvider 动态更新语句.....	83
3.5.11	@DeleteProvider 动态删除语句.....	84
3.6	使用 MyBatis Generator 自动创建代码.....	85
	本章小结.....	86
	练习题.....	86
第四章 Spring 核心技术.....		90
4.1	Spring 简介.....	90
4.1.1	Spring 的核心模块.....	90
4.1.2	Spring 框架的优势.....	91
4.1.3	Spring 开发环境的搭建.....	92
4.2	控制反转(IoC).....	93
4.2.1	IoC 的基本概念.....	93
4.2.2	依赖注入的类型.....	97

4.3	Bean 的装配	100
4.3.1	Spring 装配 Bean 的方案	100
4.3.2	Spring IoC 容器	100
4.3.3	基于注解的 Bean 装配	101
4.4	面向切面编程(AOP)	104
4.4.1	面向切面编程简介	104
4.4.2	通过切点选择连接点	106
4.4.3	使用注解创建切面	108
4.4.4	在 XML 中声明切面	112
4.5	Spring 的事务管理	116
4.5.1	事务的特性	116
4.5.2	核心接口	116
4.5.3	基本事务属性	118
4.5.3	事务状态	121
4.5.4	声明事务管理实例	121
	本章小结	123
	练习题	124
第五章 SpringMVC		127
5.1	SpringMVC 概述	127
5.2	创建第一个 SpringMVC 程序	128
5.2.1	新建项目	129
5.2.2	导入 jar 包	129
5.2.3	在 web.xml 中添加 SpringMVC 的配置	130
5.2.4	在类路径下添加 SpringMVC 的配置	130
5.2.5	建立视图文件	131
5.2.6	建立 Controller 控制层文件	132
5.2.7	部署运行项目	132
5.3	SpringMVC RequestMapping 的基本设置	133
5.4	SpringMVC 前后台数据交互	135
5.4.1	Controller 获取前台传递的参数	135
5.4.2	Controller 传递参数到前台	137
5.5	SpringMVC 文件上传和下载	138
5.5.1	文件上传	138
5.5.2	文件下载	139
5.6	SpringMVC 常用注解	141
	本章小结	145
	练习题	145

第六章 SpringMVC Spring MyBatis 集成	148
6.1 依赖包的引入	148
6.2 Spring 与 MyBatis 的集成	153
6.2.1 建立 JDBC 属性文件	154
6.2.2 建立 Spring 上下文配置文件	154
6.2.3 Log4J 的配置	157
6.2.4 JUnit 测试	157
6.3 集成 SpringMVC	161
6.3.1 建立 SpringMVC 配置文件	161
6.3.2 配置 web.xml 文件	162
6.3.3 测试	165
本章小结	166
练习题	166
第七章 项目实战	167
7.1 项目的需求分析	167
7.1.1 基础信息模块	167
7.1.2 角色管理模块	168
7.1.3 管理员管理模块	169
7.1.4 资费管理模块	169
7.1.5 账务账号管理模块	170
7.1.6 业务账号管理模块	171
7.1.7 账单管理模块	172
7.1.8 报表模块	172
7.2 概要设计	173
7.2.1 系统流程	173
7.2.2 功能模块图	173
7.3 数据库设计	174
7.3.1 数据模型	174
7.3.2 数据字典	174
7.4 功能实现	177
7.4.1 基础信息模块实现	177
7.4.2 角色管理功能实现	181
7.4.3 管理员管理功能实现	188
7.4.4 资费管理功能实现	194
7.4.5 账务账号管理功能实现	200
7.4.6 业务账号管理功能实现	208
本章小结	214

第一章 MyBatis 开发入门

欢迎您来到 MyBatis 的世界，MyBatis 是一个支持普通 SQL 查询、存储过程和高级映射的优秀持久层框架。MyBatis 消除了几乎所有的 JDBC 代码和参数的手工设置以及对结果集的检索封装。MyBatis 可以使用简单的 XML 或注解来配置和进行原始映射，将接口和 Java 的 POJOs(Plain Old Java Objects，普通的 Java 对象)映射成数据库中的记录。

本章知识要点

- MyBatis 简介
- MyBatis 的优势
- 认识第一个 MyBatis 程序
- MyBatis 日志

1.1 MyBatis 简介

MyBatis 本是 apache 的一个开源项目 iBatis，2010 年这个项目由 apache software foundation 迁移到了 google code，并且改名为 MyBatis。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)。

MyBatis 的功能架构分为三层，如图 1-1 所示。

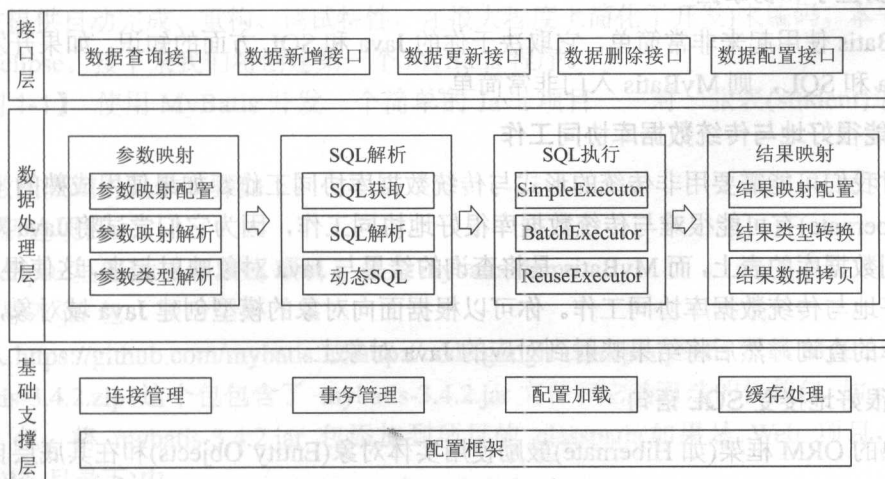


图 1-1 MyBatis 的功能架构

(1) 接口层：提供给外部使用的接口 API，开发人员通过这些本地 API 来操纵数据库。接口层一接收到调用请求就会调用数据处理层来完成具体的数据处理。

(2) 数据处理层：负责具体的参数映射、SQL 解析、SQL 执行和执行结果映射处理等。它主要的目的是根据调用的请求完成一次数据库操作。

(3) 基础支撑层：负责最基础的功能支撑，包括连接管理、事务管理、配置加载和缓存处理，这些共用的功能，被抽取出来作为最基础的组件，为上层的数据处理层提供最基础的支撑。

MyBatis 应用程序主要使用 `sqlSessionFactory` 实例，一个 `sqlSessionFactory` 实例可以通过 `sqlSessionFactoryBuilder` 获得。`sqlSessionFactoryBuilder` 可以从一个 XML 配置文件或者一个预定义的配置类的实例获得。

1.2 MyBatis 的优势

1. 消除了大量的 JDBC 冗余代码

Java 通过 Java 数据库连接(Java Data Base Connectivity, JDBC)API 来操作关系型数据库，但是 JDBC 是一个非常底层的 API，我们需要书写大量的代码来完成对数据库的操作。MyBatis 通过简单的方式实现了和 JDBC 相同的功能，如准备需要被执行的 sql Statement 对象并且将 Java 对象作为输入数据传递给 statement 对象的任务，开发人员可以专注于真正重要的方面。另外，MyBatis 使将输入的 Java 对象中的属性设置成查询参数和从 SQL 结果集上生成 Java 对象这两个过程自动化。MyBatis 还提供了其他的一些特性使持久化逻辑的实现变得简单：

- (1) 它支持复杂的 SQL 结果集数据映射到嵌套对象图结构；
- (2) 它支持一对一和一对多的结果集和 Java 对象的映射；
- (3) 它支持根据输入的数据构建动态的 SQL 语句。

2. 易上手和易掌握

MyBatis 使用起来非常简单，它取决于你的 Java 和 SQL 方面的知识。如果开发人员很熟悉 Java 和 SQL，则 MyBatis 入门非常简单。

3. 能很好地与传统数据库协同工作

有时我们可能需要用非传统的形式与传统数据库协同工作，如果使用成熟的 ORM 框架(如 Hibernate)有可能很难与传统数据库很好地协同工作，因为它们尝试将 Java 对象静态地映射到数据库的表上，而 MyBatis 是将查询的结果与 Java 对象映射起来，这使得 MyBatis 可以很好地与传统数据库协同工作。你可以根据面向对象的模型创建 Java 域对象，执行传统数据库的查询，然后将结果映射到对应的 Java 对象上。

4. 很好地接受 SQL 语句

成熟的 ORM 框架(如 Hibernate)鼓励使用实体对象(Entity Objects)和在其底层自动产生 SQL 语句。由于这种 SQL 的生成方式，使我们可能无法利用到数据库的一些特性。Hibernate 允许执行本地 SQL，但是这样会打破持久层和数据库独立的原则。

MyBatis 框架接受 SQL 语句，而不是将其对开发人员隐藏起来。由于 MyBatis 不会产生任何 SQL 语句，所以开发人员要自己准备 SQL 语句，这样就可以充分利用数据库特有的特性并且可以准备自定义的查询。另外，MyBatis 对存储过程也提供支持。

5. 提供与第三方缓存类库的集成支持

MyBatis 有内建的 sqlSession 级别的缓存机制，可缓存 select 语句的查询结果。除此之外，MyBatis 还提供与多种第三方缓存类库的集成支持，如 EHCACHE、OSCache、Hazelcast 等。

6. 引入了更好的性能

性能是关乎软件应用成功与否的关键因素之一。为了获得更好的性能，需要考虑很多方面，而对很多应用而言，数据持久化层是整个系统性能的关键。

MyBatis 支持数据库连接池，消除了为每一个请求创建一个数据库连接的开销。

MyBatis 提供内建的缓存机制，在 sqlSession 级别提供对 SQL 查询结果的缓存。即：如果你调用了相同的 select 查询，MyBatis 会将放在缓存的结果返回，而不会再去查询数据库。

MyBatis 框架并没有大量地使用代理机制，因此对于其他的过度使用代理的 ORM 框架而言，MyBatis 可以获得更好的性能。

1.3 认识第一个 MyBatis 程序

在本节开始之前假设你的系统上已经安装了 JDK1.6+(下载网站：<http://www.java.com>)和 MySQL5(下载网站：<http://www.mysql.com>)。JDK 和 MySQL 的安装过程不在本书的叙述范围之内，所以这里不再阐述。

目前 MyBatis 的最新版本是 MyBatis 3.4.2，本书将使用 MyBatis 3.4.2 版本。

本书并不限定你使用什么类型的 IDE(如 Eclipse、NetBeans IDE 或者 IntelliJ IDEA IDE，它们通过提供自动完成、重构、调试特性，在很大程度上简化了开发)来编码。本书采用的 IDE 是 Eclipse。接下来我们将创建第一个 MyBatis 程序。

【例 1-1】 使用 MyBatis 开发一个简单的 Java 项目——对一张表(student)进行数据查询。

➤ 数据查询的具体步骤如下：

(1) 新建项目。

在 eclipse 里创建一个 Java 项目，名为 mybatis-demo。

(2) 导入 jar 包。

进入 <https://github.com/mybatis> 或 <http://code.google.com/p/mybatis> 下载 MyBatis 的发布包 mybatis-3.4.2.zip。这个包包包含了 mybatis-3.4.2.jar 文件和它的可选的依赖包，如 slf4j/log4j 日志 jar 包，将 mybatis-3.4.2.jar 包添加到项目的 classpath(如果是 Web 项目，可放在 Web-INF\lib 目录下)中。

进入 <http://www.mysql.com/products/connector/> 下载 mysql 的 Java 驱动包 mysql-

connector-java-5.1.22.jar 并将其添加到项目 classpath 中。

如果项目正在使用 maven, 那么配置这些 jar 包的依赖就变得简单多了。在 pom.xml 中添加以下依赖即可:

```
<dependencies>
<!-- MyBatis 依赖 -->
<dependency>
<groupId>org.mybatis</groupId>
<artifactId>mybatis</artifactId>
<version>3.4.2</version>
</dependency>
<!-- mysql 的 java 驱动包依赖 -->
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<version>5.1.22</version>
</dependency>
</dependencies>
```

更多 maven 的相关信息请参照 <http://maven.apache.org/> 学习。

(3) 在 MySQL 里新建表并插入样本数据。

使用 SQL 脚本生成 student 表, 并且插入三条样本数据, 脚本如下:

```
-- 建表
CREATE TABLE 'student' (
'stuid' int(10) NOT NULL AUTO_INCREMENT,
'stuName' varchar(50) DEFAULT NULL,
PRIMARY KEY ('stuid')
) ENGINE = InnoDB AUTO_INCREMENT = 1 DEFAULT CHARSET = utf8;
-- 插入三条样本数据
INSERT INTO 'student' VALUES ('1', 'zhangsang');
INSERT INTO 'student' VALUES ('2', 'lisi');
INSERT INTO 'student' VALUES ('3', 'wangwu');
```

(4) 从 XML 映射 SQL 语句。

在 classpath(如果是 Web 项目, classpath 在 WEB-INF\classes 目录下)下 com.test.mapper 包里新建 StudentMapper.xml 文件, 将 SQL 信息进行映射, 其配置内容如下:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!DOCTYPE mapper
PUBLIC "-//mybatis.org/DTD Mapper 3.0/EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace = "com.test.mapper.StudentMapper">
```

```
<!-- sql查询语句映射 -->
<select id = "selectStudent" resultType = "hashmap">
SELECT * FROM student
</select>
</mapper>
```

这里的 `com.test.mapper` 是 `StudentMapper.xml` 所在的包名称。

(5) 从 XML 中创建 `sqlSessionFactory` 实例。

在 `classpath` 下新建 `mybatis-config.xml` 文件, 在文件中配置 `SqlSessionFactory` 实例信息, 其配置内容如下:

```
<?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>
<environments default = "development">
<environment id = "development">
<transactionManager type = "JDBC"/>
<!-- 数据库基本信息 -->
<dataSource type = "POOLED">
<property name = "driver" value = "com.mysql.jdbc.Driver"/>
<property name = "url" value = "jdbc:mysql:///mydb"/>
<property name = "username" value = "root"/>
<property name = "password" value = "root"/>
</dataSource>
</environment>
</environments>
<mappers>
<mapper resource = "com/test/mapper/StudentMapper.xml"/>
</mappers>
</configuration>
```

(6) 获取 `sqlSession` 并执行程序。

新建 `Test.java` 类代码如下:

```
import java.io.IOException;
import java.io.InputStream;
import java.util.List;
import java.util.Map;
import org.apache.ibatis.io.Resources;
```



```

import org.apache.ibatis.session.SqlSession;
import org.apache.ibatis.session.SqlSessionFactory;
import org.apache.ibatis.session.SqlSessionFactoryBuilder;
public class Test {
    public static void main(String[] args) {
        //类路径下配置文件名称
        String resource = "mybatis-config.xml";
        InputStream inputStream;
        SqlSession sqlSession = null ;
        try {
            //配置文件加载
            inputStream = Resources.getResourceAsStream(resource);
            //根据配置文件生成SqlSessionFactory对象
            SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);
            //sqlSession获取
            sqlSession = sqlSessionFactory.openSession();
            //执行查询请求
            List<Map> list = sqlSession.selectList("com.test.mapper.StudentMapper.selectStudent");
            //输出查询结果
            for(Map map:list)
            {
                System.out.println(map);
            }
        } catch (IOException e)
        {
            e.printStackTrace();
        }finally
        {
            sqlSession.close();
        }
    }
}

```

运行后其控制台输出结果如下：

```

{stuId = 1, stuName = zhangsan}
{stuId = 2, stuName = lisi}
{stuId = 3, stuName = wangwu}

```

(7) MyBatis 的工作机制。

MyBatis 的工作机制如图 1-2 所示。