



MyBatis 3 源码深度解析

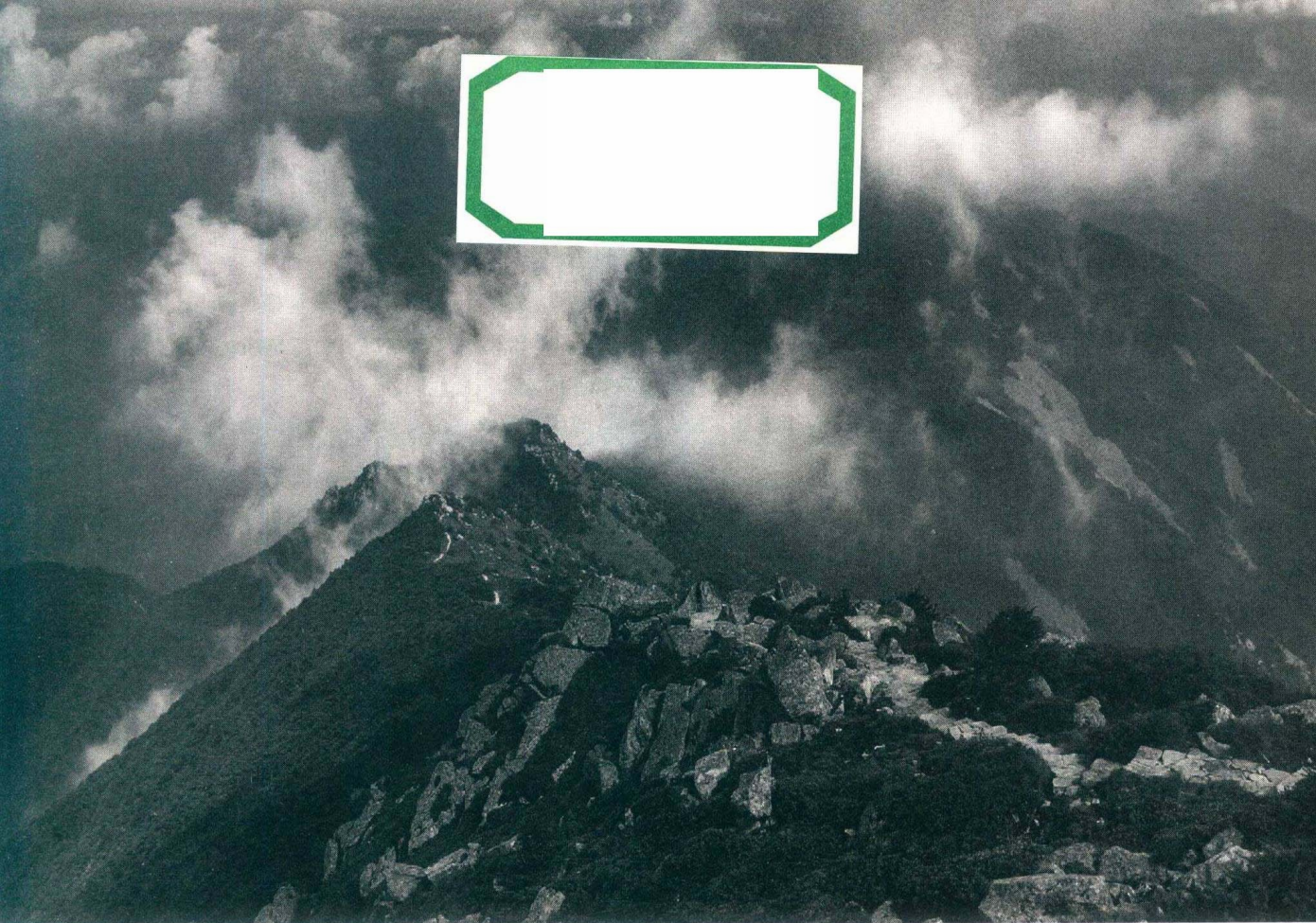
- **追本溯源**：从MyBatis的基本应用出发，挖掘底层实现，深入理解MyBatis框架
- **问题驱动**：针对现象提出疑问，然后带着问题从源码中探究答案
- **面面俱到**：涵盖MyBatis基础知识点，分析底层实现，全面掌握MyBatis

江荣波 著

 本书示例源代码



清华大学出版社



TP312
6446

MyBatis 3 源码深度解析



清华大学出版社
北京

内 容 简 介

本书从 MyBatis 源码的角度分析 Mapper 绑定过程、SqlSession 操作数据库原理、插件实现原理等，同时介绍一些 MyBatis 的高级用法，并挖掘 MyBatis 源码中使用的设计模式。

本书共 13 章，分为 MyBatis 3 源码篇和 MyBatis Spring 源码篇。第 1~11 章介绍 MyBatis 核心源码，从源码的角度分析 MyBatis 的实现原理，并介绍一些 MyBatis 的高级用法。MyBatis 大多数情况下会与 Spring 整合使用，第 12~13 章介绍 MyBatis Spring 的实现原理，并分析 MyBatis Spring 模块的核心代码。

本书适合掌握了 MyBatis 的基本用法并希望了解 MyBatis 底层实现的 Java 开发人员、架构师以及对 Java 开源项目感兴趣的读者阅读。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目 (CIP) 数据

MyBatis 3 源码深度解析/江荣波著. —北京：清华大学出版社，2019
ISBN 978-7-302-53561-4

I. ①M… II. ①江… III. ①JAVA 语言—程序设计 IV. ①TP312.8

中国版本图书馆 CIP 数据核字 (2019) 第 179900 号

责任编辑：夏毓彦

封面设计：王翔

责任校对：闫秀华

责任印制：沈露

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦 A 座

邮 编：100084

社总机：010-62770175

邮 购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者：北京国马印刷厂

经 销：全国新华书店

开 本：190mm×260mm

印 张：16.5

字 数：423 千字

版 次：2019 年 10 月第 1 版

印 次：2019 年 10 月第 1 次印刷

定 价：69.00 元

产品编号：081258-01

前 言

在写作本书时，SSM (Spring、Spring MVC 和 MyBatis) 框架已经成为很多互联网公司的标配。笔者最早接触 MyBatis 框架是在 2014 年，当时接手的是一个非常成熟的网上银行系统，项目中使用的持久层框架就是 MyBatis 的前身——iBatis 框架。后来换了两份工作，新的公司也都选择 MyBatis 作为持久层框架。从 iBatis 过渡到 MyBatis 框架几乎不需要任何学习成本，MyBatis 延续了 iBatis 简单易用的特点，优化了 SQL 配置方式，引用 OGNL 表达式来支持动态 SQL 配置，使得动态 SQL 配置更加优雅，而且更容易理解。在笔者看来，MyBatis 相对于 iBatis 框架最大的创新是引入了 SQL Mapper 的概念。我们可以将 XML 文件中的 SQL 配置与一个 Java 接口进行绑定，SQL 配置的命名空间对应 Java 接口的完全限定名，而具体的每个 SQL 语句的配置对应 Java 接口中的一个方法，建立绑定后，可以通过调用 Java 接口中定义的方法来执行 XML 文件中配置的 SQL 语句。

MyBatis 作为持久层框架，以其小巧轻便、SQL 可配置、使用简单等特点深受广大 Java 开发者喜爱。然而大多数开发人员对 MyBatis 框架的理解仅局限于使用，并不理解框架底层的实现原理。作为一名开发人员，阅读开源框架的源码，可以学习源码中对设计模式及面向对象设计原则的应用，有助于提升自身编码能力。笔者在工作之余，深入研究了 MyBatis 框架的源码，本书将会从源码的角度分析 MyBatis 框架各个特性的实现原理。

阅读准备

在阅读本书之前，读者需要准备如下开发环境：

- JDK1.8 或以上版本
- Apache Maven 构建工具
- IntelliJ IDEA 开发工具
- Git 版本控制工具

内容概要

本书主要分为两篇：第 1 篇为 MyBatis 3 源码篇（第 1~11 章），主要介绍 MyBatis 框架各个特性的源码实现；第 2 篇章为 MyBatis Spring 源码篇（第 12~13 章），主要介绍 MyBatis 框架与 Spring 框架整合的原理及 MyBatis Spring 模块的实现细节。下面是本书的内容大纲。

第 1 章 搭建 MyBatis 源码环境

主要介绍如何搭建 MyBatis 源码调试环境，包括 MyBatis 框架源码获取途径、如何导入集成开发工具以及如何运行 MyBatis 源码中的测试用例。

第 2 章 JDBC 规范详解

MyBatis 框架是对 JDBC 轻量级的封装，熟练掌握 JDBC 规范有助于理解 MyBatis 框架实现原理。本章将详细介绍 JDBC 规范相关细节，已经全面掌握 JDBC 规范的读者可以跳过该章。

第 3 章 MyBatis 常用工具类

介绍 MyBatis 框架中常用的工具类，避免读者因对这些工具类的使用不熟悉而导致对框架主流程理解的干扰，这些工具类包括 MetaObject、ObjectFactory、ProxyFactory 等。

第 4 章 MyBatis 核心组件介绍

介绍 MyBatis 的核心组件，包括 Configuration、SqlSession、Executor、MappedStatement 等，包括这些组件的作用及 MyBatis 执行 SQL 语句的核心流程。

第 5 章 SqlSession 的创建过程

主要介绍 SqlSession 组件的创建过程，包括 MyBatis 框架对 XPath 方式解析 XML 封装的工具类、MyBatis 主配置文件解析生成 Configuration 对象的过程。

第 6 章 SqlSession 执行 Mapper 过程

本章介绍 Mapper 接口注册的过程、SQL 配置转换为 MappedStatement 对象并注册到 Configuration 对象的过程。除此之外，本章还将介绍 SqlSession 对象执行 Mapper 的过程。

第 7 章 MyBatis 缓存

本章首先介绍 MyBatis 一级缓存和二级缓存的使用细节，接着介绍一级缓存和二级缓存的实现原理，最后介绍 MyBatis 如何整合 Redis 作为二级缓存。

第 8 章 MyBatis 日志实现

基于 Java 语言的日志框架比较多，比较常用的有 Logback、Log4j 等，本章介绍 Java 的日志框架发展史，并介绍这些日志框架之间的关系，最后介绍 MyBatis 自动查找日志框架的实现原理。

第 9 章 动态 SQL 实现原理

本章主要介绍 MyBatis 动态 SQL 的使用、动态 SQL 配置转换为 SqlSource 对象的过程以及动态 SQL 的解析原理，最后从源码的角度分析动态 SQL 配置中#{ } 和\${ } 参数占位符的区别。

第 10 章 MyBatis 插件原理及应用

本章介绍 MyBatis 插件的实现原理，并以实际的案例介绍如何自定义 MyBatis 插件。在本章中将会实现两个 MyBatis 插件，分别为分页查询插件和慢 SQL 统计插件。

第 11 章 MyBatis 级联映射与懒加载

本章介绍 MyBatis 中一对一、一对多级联映射和懒加载机制的使用细节，并介绍级联映射和懒加载的源码实现。

第 12 章 MyBatis 与 Spring 整合案例

在介绍 MyBatis 框架与 Spring 整合原理之前，需要了解 MyBatis 整合 Spring 的基本配置，本章以一个用户注册 RESTful 接口案例作为 MyBatis 框架与 Spring 框架整合的最佳实践。

第 13 章 MyBatis Spring 的实现原理

首先介绍 Spring 框架中的一些核心概念和 Spring IoC 容器的启动过程，接着介绍 MyBatis 和 Spring 整合后动态代理产生的 Mapper 对象是如何与 Spring IoC 容器进行关联的，最后介绍 MyBatis 整合 Spring 事务管理的实现原理。

随书源码

本书相关源码托管在 Github 上，读者可以从 Github 仓库获取随书源码。

源码地址：<https://github.com/rongbo-j/mybatis-book>。

图书勘误

由于个人能力有限，书中可能有表述不到位或者对知识点理解欠妥的地方，欢迎读者批评指正。若有任何疑问，均可以在随书源码 Github 仓库上提交。

勘误地址：<https://github.com/rongbo-j/mybatis-book/issues>。

致谢

本书从写作到完稿用了一年多时间，这个过程对于笔者来说是一个极大的考验。作为一名程序员，只有不断地提升，才会觉得充实。把大量的时间用在写作上，时常会因为摄入新知识而恐慌，感谢在本书写作过程中家人对我精神上的支持。另外，特别感谢夏毓彦老师和清华大学出版社的工作人员，有你们的帮助才有本书的顺利出版。

著 者

2019年5月

目 录

第 1 篇 MyBatis 3 源码

第 1 章 搭建 MyBatis 源码环境	3
1.1 MyBatis 3 简介	3
1.2 环境准备	4
1.3 获取 MyBatis 源码	4
1.4 导入 MyBatis 源码到 IDE	6
1.5 HSQLDB 数据库简介	9
1.6 本章小结	11
第 2 章 JDBC 规范详解	13
2.1 JDBC API 简介	13
2.1.1 建立数据源连接	14
2.1.2 执行 SQL 语句	15
2.1.3 处理 SQL 执行结果	16
2.1.4 使用 JDBC 操作数据库	16
2.2 JDBC API 中的类与接口	17
2.2.1 java.sql 包详解	17
2.2.2 javax.sql 包详解	20
2.3 Connection 详解	24
2.3.1 JDBC 驱动类型	24
2.3.2 java.sql.Driver 接口	26
2.3.3 Java SPI 机制简介	27
2.3.4 java.sql.DriverAction 接口	29
2.3.5 java.sql.DriverManager 类	29
2.3.6 javax.sql.DataSource 接口	31
2.3.7 使用 JNDI API 增强应用的可移植性	32
2.3.8 关闭 Connection 对象	34
2.4 Statement 详解	35
2.4.1 java.sql.Statement 接口	35
2.4.2 java.sql.PreparedStatement 接口	39

2.4.3	java.sql.CallableStatement 接口	43
2.4.4	获取自增长的键值	44
2.5	ResultSet 详解	45
2.5.1	ResultSet 类型	45
2.5.2	ResultSet 并行性	46
2.5.3	ResultSet 可保持性	46
2.5.4	ResultSet 属性设置	47
2.5.5	ResultSet 游标移动	47
2.5.6	修改 ResultSet 对象	48
2.5.7	关闭 ResultSet 对象	50
2.6	DatabaseMetaData 详解	51
2.6.1	创建 DatabaseMetaData 对象	51
2.6.2	获取数据源的基本信息	51
2.6.3	获取数据源支持特性	53
2.6.4	获取数据源限制	53
2.6.5	获取 SQL 对象及属性	54
2.6.6	获取事务支持	54
2.7	JDBC 事务	54
2.7.1	事务边界与自动提交	55
2.7.2	事务隔离级别	55
2.7.3	事务中的保存点	56
2.8	本章小结	57
第 3 章	MyBatis 常用工具类	58
3.1	使用 SQL 类生成语句	58
3.2	使用 ScriptRunner 执行脚本	64
3.3	使用 SqlRunner 操作数据库	67
3.4	MetaObject 详解	71
3.5	MetaClass 详解	72
3.6	ObjectFactory 详解	73
3.7	ProxyFactory 详解	74
3.8	本章小结	75
第 4 章	MyBatis 核心组件介绍	76
4.1	使用 MyBatis 操作数据库	76
4.2	MyBatis 核心组件	80
4.3	Configuration 详解	82
4.4	Executor 详解	88
4.5	MappedStatement 详解	90

4.6	StatementHandler 详解.....	92
4.7	TypeHandler 详解	94
4.8	ParameterHandler 详解	97
4.9	ResultSetHandler 详解	98
4.10	本章小结	100
第 5 章	SqlSession 的创建过程.....	101
5.1	XPath 方式解析 XML 文件.....	101
5.2	Configuration 实例创建过程.....	104
5.3	SqlSession 实例创建过程.....	108
5.4	本章小结	109
第 6 章	SqlSession 执行 Mapper 过程.....	110
6.1	Mapper 接口的注册过程.....	110
6.2	MappedStatement 注册过程	114
6.3	Mapper 方法调用过程详解.....	119
6.4	SqlSession 执行 Mapper 过程.....	126
6.5	本章小结	130
第 7 章	MyBatis 缓存.....	131
7.1	MyBatis 缓存的使用.....	131
7.2	MyBatis 缓存实现类.....	132
7.3	MyBatis 一级缓存实现原理.....	135
7.4	MyBatis 二级缓存实现原理.....	138
7.5	MyBatis 使用 Redis 缓存.....	142
7.6	本章小结	145
第 8 章	MyBatis 日志实现	146
8.1	Java 日志体系	146
8.2	MyBatis 日志实现.....	149
8.3	本章小结	155
第 9 章	动态 SQL 实现原理	156
9.1	动态 SQL 的使用	156
9.2	SqlSource 与 BoundSql 详解	159
9.3	LanguageDriver 详解	161
9.4	SqlNode 详解	164
9.5	动态 SQL 解析过程.....	169
9.6	从源码角度分析#{ }和\${ }的区别.....	179
9.7	本章小结	182

第 10 章 MyBatis 插件原理及应用.....	184
10.1 MyBatis 插件实现原理.....	184
10.2 自定义一个分页插件.....	193
10.3 自定义慢 SQL 统计插件.....	198
10.4 本章小结.....	200
第 11 章 MyBatis 级联映射与懒加载.....	201
11.1 MyBatis 级联映射详解.....	201
11.1.1 准备工作.....	201
11.1.2 一对多关联映射.....	205
11.1.3 一对一关联映射.....	206
11.1.4 Discriminator 详解.....	209
11.2 MyBatis 懒加载机制.....	210
11.3 MyBatis 级联映射实现原理.....	212
11.3.1 resultMap 详解.....	212
11.3.2 resultMap 解析过程.....	213
11.3.3 级联映射实现原理.....	218
11.4 懒加载实现原理.....	225
11.5 本章小结.....	230

第 2 篇 MyBatis Spring 源码

第 12 章 MyBatis 与 Spring 整合案例.....	233
12.1 准备工作.....	233
12.2 MyBatis 与 Spring 整合.....	234
12.3 用户注册案例.....	236
12.4 本章小结.....	239
第 13 章 MyBatis Spring 的实现原理.....	240
13.1 Spring 中的一些概念.....	240
13.2 Spring 容器启动过程.....	243
13.3 Mapper 动态代理对象注册过程.....	244
13.4 MyBatis 整合 Spring 事务管理.....	248
13.5 本章小结.....	253

第 1 篇 MyBatis 3 源码

第 1 章

搭建 MyBatis 源码环境

1.1 MyBatis 3 简介

MyBatis 源于 Apache 的一个开源项目 iBatis。2002 年，Clinton Begin 开发了 iBatis 框架，并引入了 SQL 映射作为持久化层开发的一种方法，不久后 Clinton Begin 将 iBatis 捐献给 Apache 软件基金会。2010 年，这个项目由 Apache 迁移到了 Google Code，并改名为 MyBatis。2013 年 11 月，MyBatis 迁移到目前最大的源代码托管平台 Github。

MyBatis 是一款在持久层使用的 SQL 映射框架，可以将 SQL 语句单独写在 XML 配置文件中，或者使用带有注解的 Mapper 映射类来完成数据库记录到 Java 实体的映射。与另一款主流的 ORM 框架 Hibernate 不同，MyBatis 属于半自动的 ORM 框架，它虽然不能将不同数据库的影响隔离开，仍然需要自己编写 SQL 语句，但是可以灵活地控制 SQL 语句的构造，将 SQL 语句的编写和程序的运行分离开，使用更加便捷。

目前，Java 实现的持久化框架比较多，名气相对较大的有 Hibernate、Speedment、Spring Data JPA、ActiveJPA 等。总结一下，MyBatis 能够流行起来的主要原因有以下几点：

- (1) 消除了大量的 JDBC 冗余代码，包括参数设置、结果集封装等。
- (2) SQL 语句可控制，方便查询优化，使用更加灵活。
- (3) 学习成本比较低，对于新用户能够快速学习使用。
- (4) 提供了与主流 IoC 框架 Spring 的集成支持。
- (5) 引入缓存机制，提供了与第三方缓存类库的集成支持。

MyBatis 这些优秀的特性使它成为目前最受欢迎的 ORM 框架之一。读者在阅读本书时可能已经熟练掌握了 MyBatis 的基本使用，本书将从源码的角度介绍 MyBatis 框架的底层实现。

1.2 环境准备

搭建 MyBatis 源码调试环境，我们首先需要安装 JDK、Maven、Eclipse（或 IntelliJ IDEA）、Git、MySQL（可选）等常用工具。需要注意的是，mybatis-spring 源码中用到了 Java 8 的新特性，例如 Lambda 表达式、Streams API 等，所以 JDK 的版本必须是 1.8 以上。MyBatis 源码使用 Maven 作为依赖管理和项目构建工具，我们需要安装 Maven 构建工具。另外，目前 MyBatis 源码托管在 Github 上，我们需要使用 Git 从远程仓库获取源码。最后，读者还需要安装一款自己比较熟悉的集成开发工具，例如 Eclipse 或者 IntelliJ IDEA 等。

注 意

JDK1.8 下载地址：<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>。

Maven 地址：<http://maven.apache.org/download.cgi>。

Git 下载地址：<https://git-scm.com/>。

1.3 获取 MyBatis 源码

1.2 节介绍了搭建 MyBatis 源码调试环境需要的工具，这些工具的安装比较简单，而且本书面向的读者为 Java 开发人员，相信大部分读者的机器上本身就具备这样的环境。如果缺少这些工具，读者可以自行安装。准备工作完成后，我们就可以获取 MyBatis 的源码了。MyBatis 源码目前托管在 Github 上，源码地址为 <https://github.com/mybatis/mybatis-3>。

MyBatis 框架在 Github 上的仓库如图 1-1 所示，如果读者想为 MyBatis 项目贡献源码，可以注册 Github 账户，然后单击 Fork 按钮，在自己的仓库中创建 MyBatis 项目的副本，代码开发测试完毕后，向上游仓库提交 Pull Request 即可。当 MyBatis 源码维护者将我们提交的代码合并后，我们就可以成为 MyBatis 源码贡献者。关于 Github 的 Fork + Pull Request 工作流程模式这里不做详细介绍，有兴趣的读者可以参考 Github 官方文档。

回归主题，我们的目的是获取 MyBatis 的源码，读者可以先单击图 1-1 中的 Clone or download 按钮再单击 Download Zip 按钮直接下载源码的压缩包。除了这种方式外，我们还可以使用 Git 客户端克隆一份代码到本地，具体操作如下：

打开 Git Bash 控制台，执行 git clone 命令：

```
git clone https://github.com/mybatis/spring.git
```

上面的命令执行结束后，MyBatis 源码项目就会克隆到本地。本书除了介绍 MyBatis 源码外，还会详细介绍 MyBatis 与 Spring 进行整合的原理，因此我们还需要获取 mybatis-spring 项目的源码，该项目同样托管在 Github 上，源码地址为 <https://github.com/mybatis/spring>。

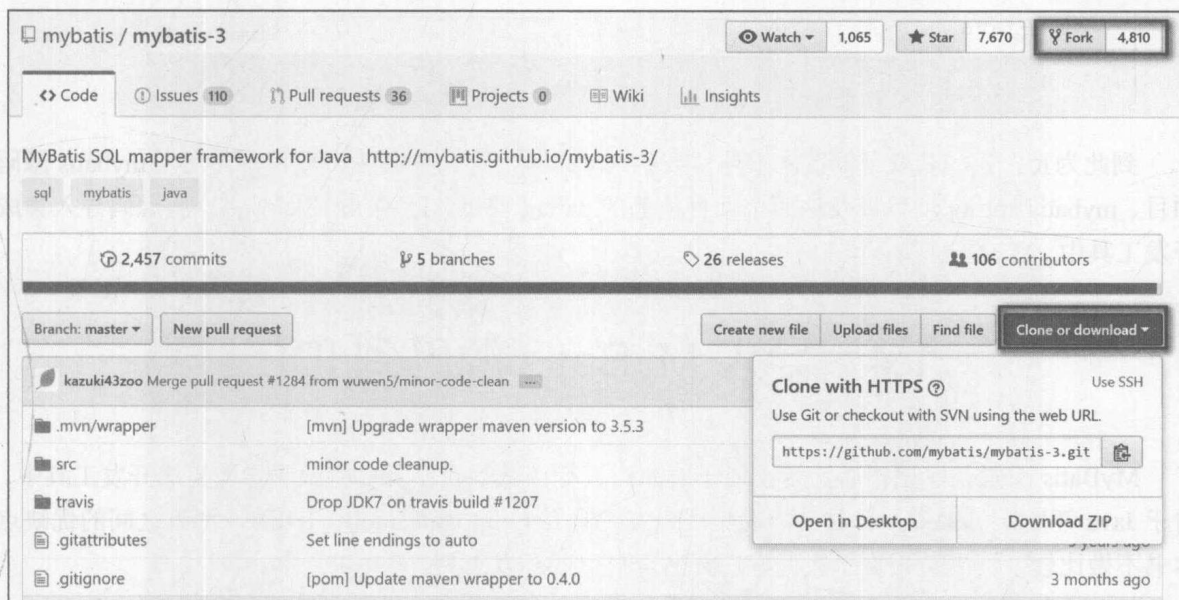


图 1-1 MyBatis 源码 Github 仓库

我们同样可以使用 `git clone` 命令在本地克隆一份 `mybatis-spring` 项目源码，具体命令如下：

```
git clone https://github.com/mybatis/parent.git
```

需要注意的是，MyBatis 源码项目使用 Maven 作为项目构建工具，`mybatis` 和 `mybatis-spring` 项目都依赖于一个公共的 `parent` 项目，该项目中没有任何代码，只是定义了一些公共的属性及项目依赖的插件信息，我们还需要把 `mybatis-parent` 项目（地址为 `https://github.com/mybatis/parent`）克隆到本地。

我们依然使用 `git clone` 命令将 `mybatis-parent` 项目克隆到本地，具体如下：

```
git clone https://github.com/mybatis/parent.git
```

3 个项目全部克隆到本地后，需要放在同一个目录下。源码目录结构如下：

```

├── mybatis-3
│   ├── src
│   │   ├── main
│   │   │   └── java
│   │   └── ...
│   └── test
│       └── java
│           └── ...
├── parent
│   └── src
├── spring
│   ├── src
│   │   ├── main
│   │   │   └── java
│   └── ...
└── test

```

```
| java  
| ...  
| resources
```

到此为止，我们获取了阅读本书所需要的 MyBatis 源码，一共 3 个项目，分别为 mybatis 源码项目、mybatis-spring 项目以及这两个项目依赖的 parent 项目。1.4 节介绍如何将这些源码导入集成开发工具中。

1.4 导入 MyBatis 源码到 IDE

MyBatis 源码获取完毕后，为了便于对源码进行调试，我们需要将源码导入集成开发工具中。对于 Java 开发人员来说，目前主流的集成开发工具有 Eclipse 和 IntelliJ IDEA。两者之间的优缺点本书不做比较，读者可根据个人偏好选择适合自己的开发工具。

接下来笔者以 IntelliJ IDEA 开发工具为例介绍如何将 MyBatis 源码导入 IDEA 中。

首先打开 IntelliJ IDEA 开发工具，单击 File→New→Project 菜单，如图 1-2 所示。

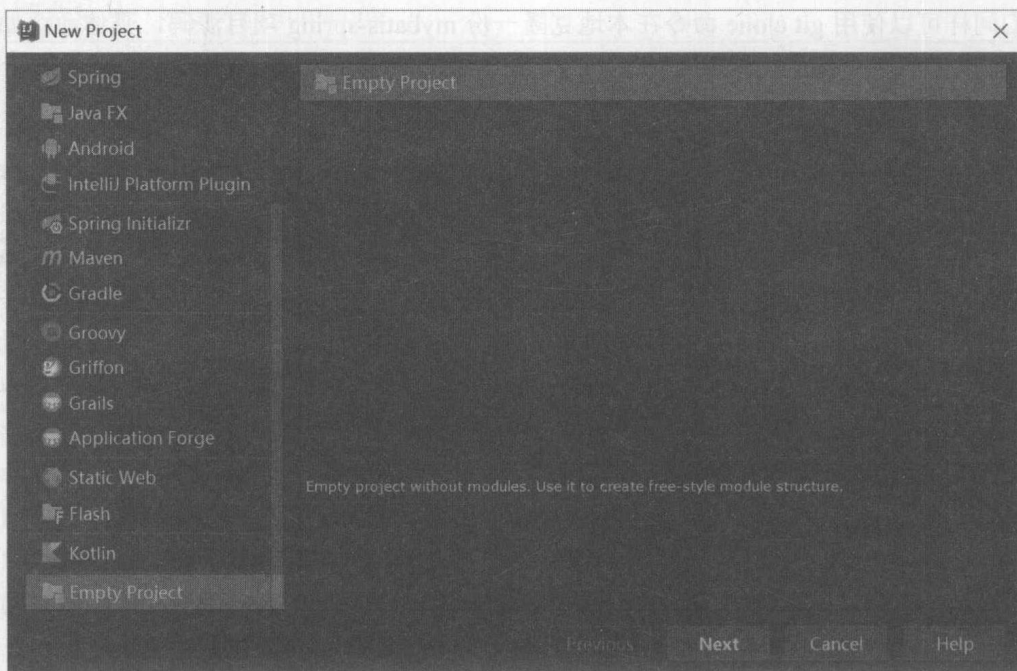


图 1-2 IntelliJ IDEA 开发工具新建项目

单击 Next 按钮，进入下一步，输入项目名称和项目路径，如图 1-3 所示。然后单击 Finish 按钮完成空项目的创建。



图 1-3 IDEA 新建项目对话框

到此为止，一个空项目已经创建完毕。接下来我们需要将 MyBatis 源码以模块（Module）的形式导入 IntelliJ IDEA 开发开发工具中，具体步骤如下：

单击 File→New→Module from Existing Sources...菜单，在对话框中选择 MyBatis 源码所在的路径，然后单击 OK 按钮，如图 1-4 所示。

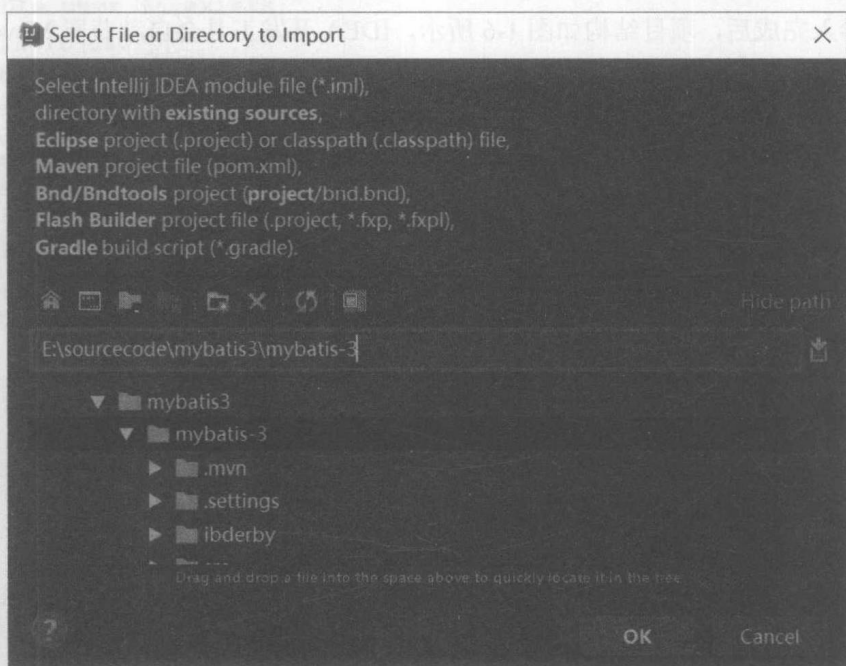


图 1-4 IDEA 导入 Module 对话框

由于 MyBatis 源码项目使用 Maven 作为构建工具，因此我们需要在如图 1-5 所示的对话框中选择 Maven 选项。然后单击 Next 按钮，在后面的对话框中，我们可以保持默认的选项，一直单击 Next 按钮。最后单击 Finish 按钮即可完成 MyBatis 源码项目的导入。