



# 咕泡学院 JAVA VIP 高级课程教案

## MYBATIS 源码分析系列专题

### 第一课 MyBatis 介绍及使用（实用篇）

主讲	James QQ 2904270631
版本	V1.0
适用对象	咕泡学院 JAVA 高级 VIP 学员

本文档版权归咕泡学院所有，禁止未经允许私自传播。

---

## 目录

咕泡学院 Java VIP 高级课程教案.....	0
MyBatis 源码分析系列专题.....	0
第一课 MyBatis 介绍及使用（实用篇） .....	0
MyBatis 介绍及使用 .....	2
1. 认识 MyBatis.....	2
2. 使用 MyBatis.....	4
3. XML 映射配置文件.....	6
4. Mapper 文件认识 .....	7
5. Batch 批量操作.....	7
6. 分页.....	8
7. 联合查询.....	8
8. 作业.....	错误!未定义书签。

---

# MyBatis 介绍及使用

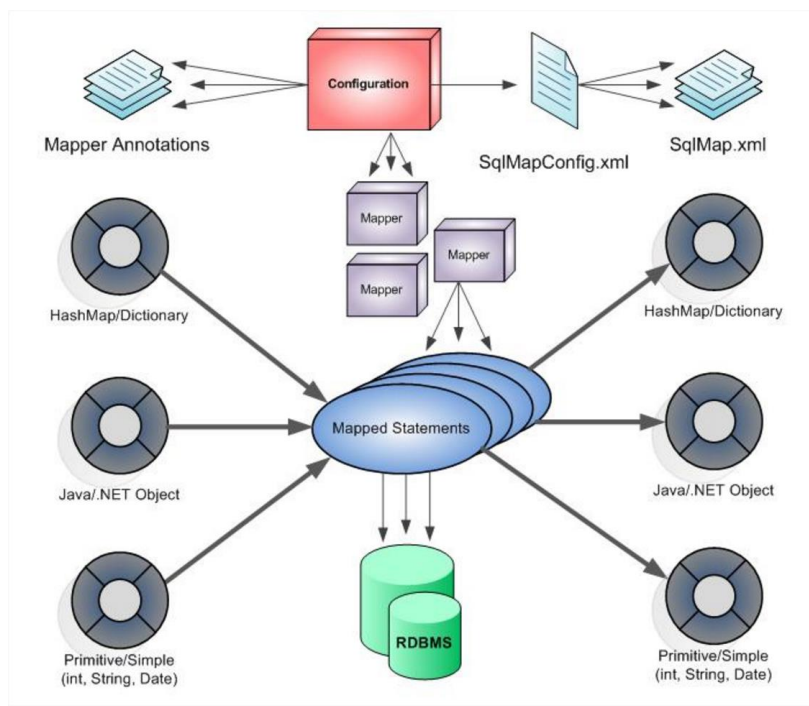
## 1. 认识 MyBatis

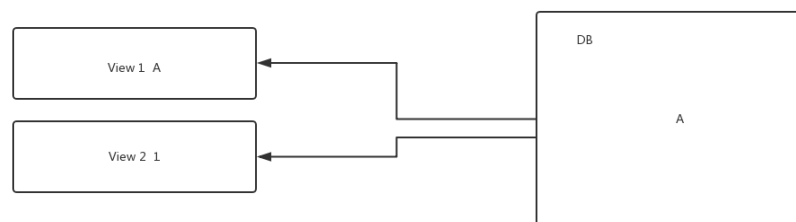
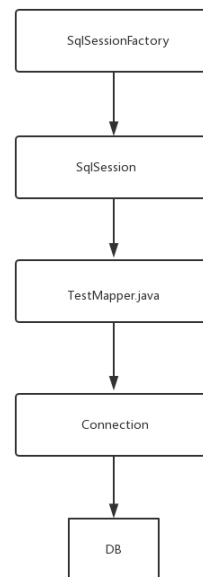
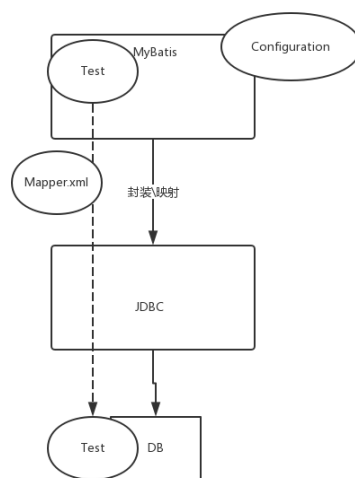
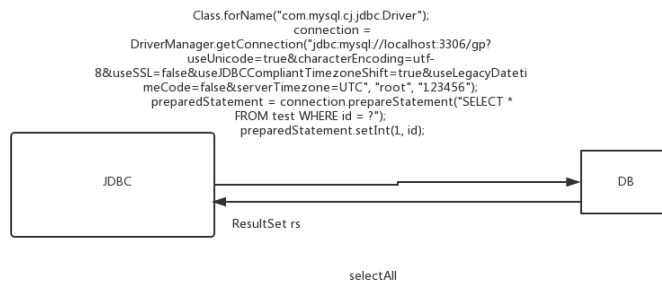
### a) What is MyBatis?

*MyBatis is a first class persistence framework with support for custom SQL, stored procedures and advanced mappings. MyBatis eliminates almost all of the JDBC code and manual setting of parameters and retrieval of results. MyBatis can use simple XML or Annotations for configuration and map primitives, Map interfaces and Java POJOs (Plain Old Java Objects) to database records.*

### b) 对比 JDBC 和 MyBatis

c)





## 2. 使用 MyBatis

### a) 使用过程

- i. 生成 bean 和配置文件
- ii. 配置 SqlSessionFactory
- iii. 使用(两种方式)

- 
1. Mapper.xml 形式
  2. Annotation 形式

	Pros	Cons
Mapper.xml	<ul style="list-style-type: none"><li>● 跟接口分离、统一管理</li><li>● 复杂的语句可以不影响接口的可读性</li></ul>	<ul style="list-style-type: none"><li>● 过多的 XML 文件</li></ul>
Annotation	<ul style="list-style-type: none"><li>● 接口就能看到 <u>sql</u> 语句，可读性高，不需要再去找 xml 文件，方便</li></ul>	<ul style="list-style-type: none"><li>● 复杂的联合查询不好维护，代码可读性差</li></ul>

iv. SqlSessionFactory\SqlSession\Mapper 推荐作用域

	Scope
SqlSessionFactoryBuilder	method
SqlSessionFactory	application
SqlSession	request/method （可以认为是线程级）
Mapper	method

- b) MyBatis Generator 使用
- i. 配置 generator 插件

```

<plugin>
  <groupId>org.mybatis.generator</groupId>
  <artifactId>mybatis-generator-maven-plugin</artifactId>
  <version>1.3.3</version>
  <configuration>
    <configurationFile>${project.basedir}/src/main/resources/mybatis/generatorCo
  </configuration>
</plugin>

```

- ii. generatorConfig.xml
- iii. 执行 mvn mybatis-generator:generate
- iv. 生成 Bean 和 Example
- v.

### 3. XML 映射配置文件

a) <http://www.mybatis.org/mybatis-3/configuration.html>

b) Type Handlers

- i. 可以 override 原有的 handlers
- ii. 可以创造自己的 handler 补充没实现的类

c) objectFactory

d) plugins

- i. 只能针对以下操作做 plugin

- Executor (update, query, flushStatements, commit, rollback, getTransaction, close, isClosed)
- ParameterHandler (getParameterObject, setParameters)
- ResultSetHandler (handleResultSets, handleOutputParameters)
- StatementHandler (prepare, parameterize, batch, update, query)

- ii. 步骤

1. 实现 org.apache.ibatis.plugin.Interceptor
2. 定义到底对谁做 plugin

```

@Intercepts({@Signature(type = Executor.class,
    method = "query",
    args = {MappedStatement.class, Object.class, RowBounds.class, ResultHandler.class})
})

```

---

### 3. Plugin 方法

```
public Object plugin(Object target) {  
    return Plugin.wrap(target, interceptor: this);  
}
```

### 4. 传参数

```
public void setProperties(Properties properties) {  
  
}
```

#### iii. 输出 MyBatis 日志

#### e) environments

**NOTE** If you are planning to use MyBatis with Spring there is no need to configure any TransactionManager because the Spring module will set its own one overriding any previously set configuration.

## 4. Mapper 文件认识

- a. namespace 关联到 mapper 接口，区分类似 package 的作用
- b. resultMap 结果映射
- c. sql SQL 复用
- d. select Insert update delete CRDU 操作
- e. 动态 SQL <http://www.mybatis.org/mybatis-3/dynamic-sql.html>

## 5. Batch 批量操作

- a. 三种方式

	性能	
--	----	--



for 循环	低 每次都要 IO	
foreach 拼 SQL (性能最高)	高	有 SQL 长度限制, 定好 List 大小
ExecutorType.BATCH		使用见代码

## 6. 分页

### a. 逻辑分页

- a) org.apache.ibatis.executor.resultset.DefaultResultSetHandler#handleRowValuesForSimpleResultMap 内存分页而已

### b. 物理分页

- a) limit 0,10
- b) 分页面插件 <https://github.com/pagehelper/Mybatis-PageHelper>

## 7. 联合查询

### a. 两种 result resultType vs resultMap

	Prons	Crons
resultType	多表关联字段是清楚知道的, 性能调优直观	创建很多实体类
resultMap	不需要写 join 语句	N+1 问题

### b. 结果映射有两种方式

- a) 最原始的方式 连接然后用 resultType/resultMap  

```
select * from blog b,author a where b.author =a.id;
```

b) MyBatis 的方式

```
<select id="selectBlogAuthor" resultMap="blogResultMap" parameterType="int">
    select
    <include refid="Base_Column_List" />
    from blog
    where bid = #{id, jdbcType=INTEGER}
</select>
<!-- 嵌套查询 end -->
```

c. 嵌套结果 嵌套查询

d. org.apache.ibatis.executor.resultset.DefaultResultSetHandler#getPropertyMappingValue

```
SELECT * FROM gp.author;CREATE TABLE `author` (
  `aid` int(11) NOT NULL AUTO_INCREMENT,
  `author_name` varchar(45) COLLATE utf8_bin DEFAULT NULL,
  PRIMARY KEY (`aid`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

```
CREATE TABLE `blog` (
  `bid` int(11) NOT NULL AUTO_INCREMENT,
  `name` varchar(45) COLLATE utf8_bin DEFAULT NULL,
  `author_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`bid`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

```
CREATE TABLE `posts` (
  `pid` int(11) NOT NULL AUTO_INCREMENT,
  `post_name` varchar(45) DEFAULT NULL,
```

---

```
`blog_id` int(11) DEFAULT NULL,  
  
PRIMARY KEY (`pid`)  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;
```



做技术人的指路明灯，职场生涯的精神导师。