



# 咕泡学院 JAVA VIP 高级课程教案

## MYBATIS 源码分析系列专题

### MyBatis 源码分析实现篇

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
MyBatis 源码分析实现篇.....	0
MyBatis 源码分析及实现 .....	2
1. 先看大架构 .....	2
2. 分析找到切入点 .....	4
3. 先来一个 1.0 版本.....	5
4. 再来一个 2.0 版本.....	5
5. mybatis-spring-x.x.x.jar .....	8
6. MyBatis VS Hibernate.....	9
7. 回顾总结.....	10

---

# MyBatis 源码分析及实现

源码分析专题设计理念

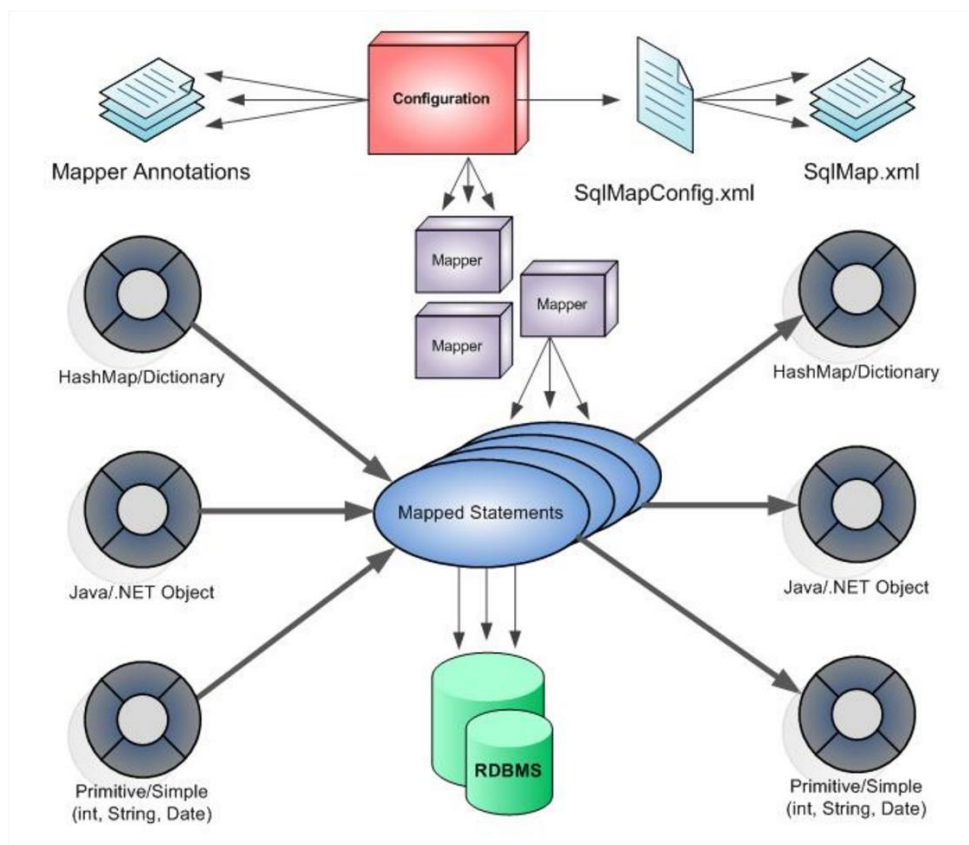
用手机（学员经验）--》

拆手机（spring 各个模块、设计模式） --》

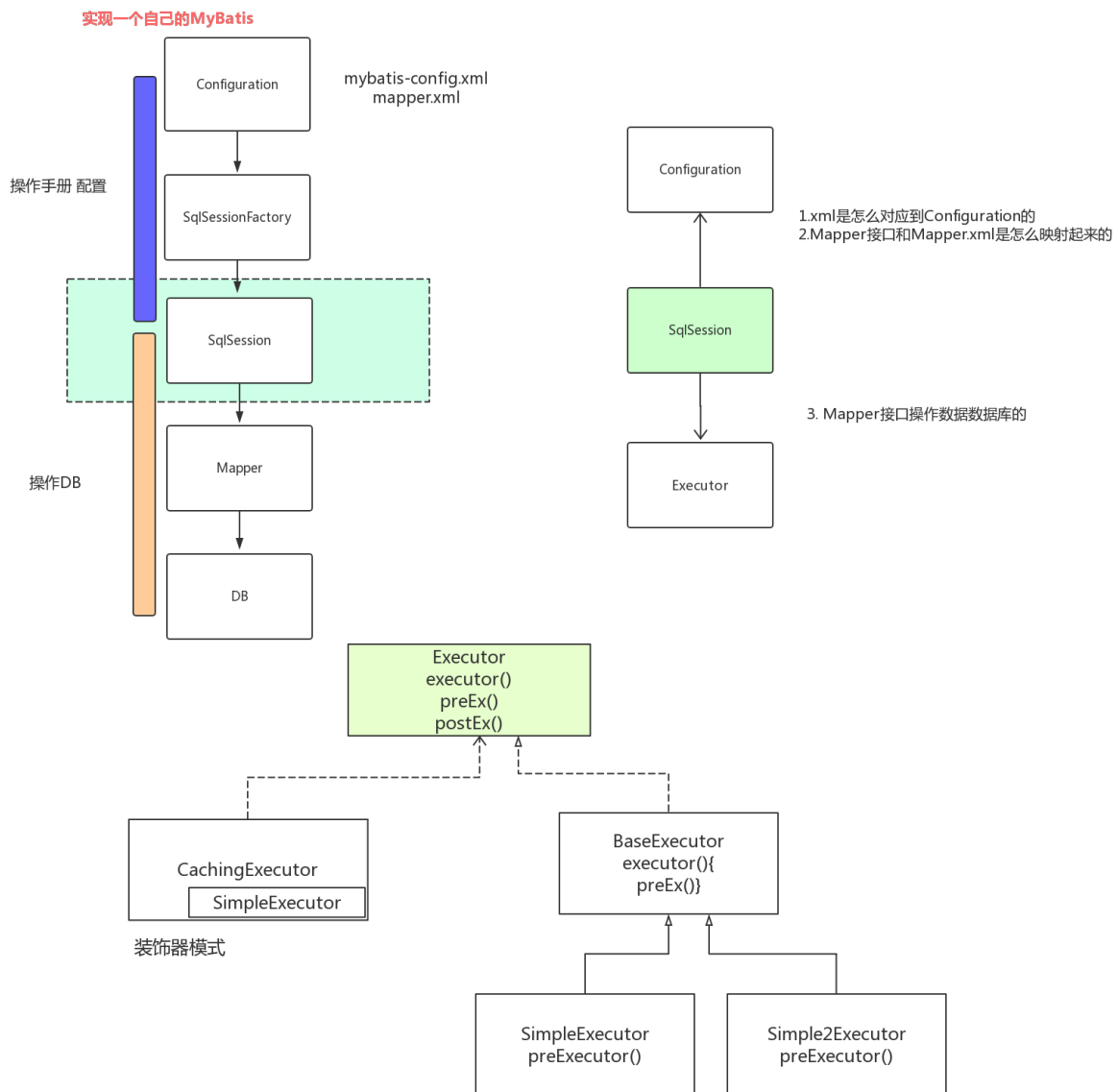
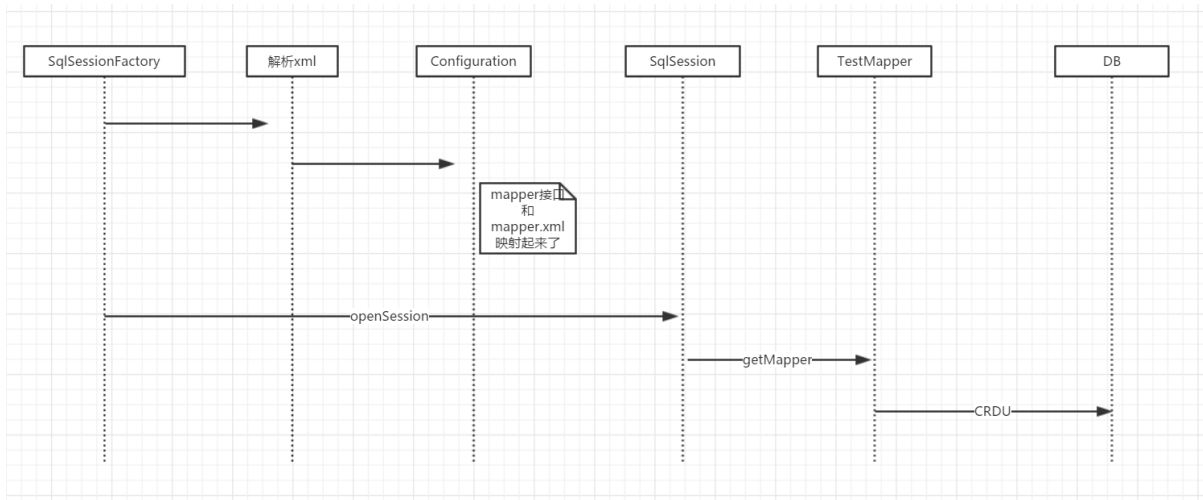
装回去（MyBatis 自己实现）

## 1. 先看大架构

a)



## 2. 分析找到切入点



---

## XML 解析成 Configuration

org.apache.ibatis.builder.xml.XMLConfigBuilder#parseConfiguration

mapper.xml 解析

org.apache.ibatis.builder.xml.XMLConfigBuilder#mapperElement

configuration.mappedStatements

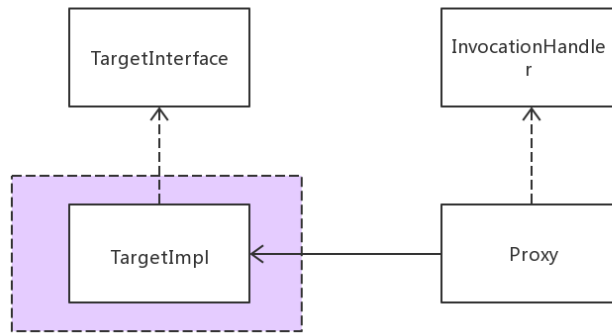
### 3. 先来一个 1.0 版本

### 4. 再来一个 2.0 版本

1.0 版本不管是抄或者自己写，一定要实现，一定要把 MyBatis 的大的架构搞清楚。

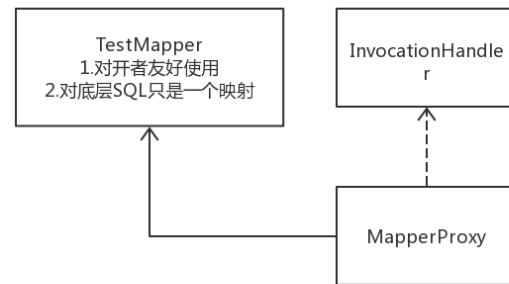
2.0 升级思考、优化 all in one → modules

动态代理

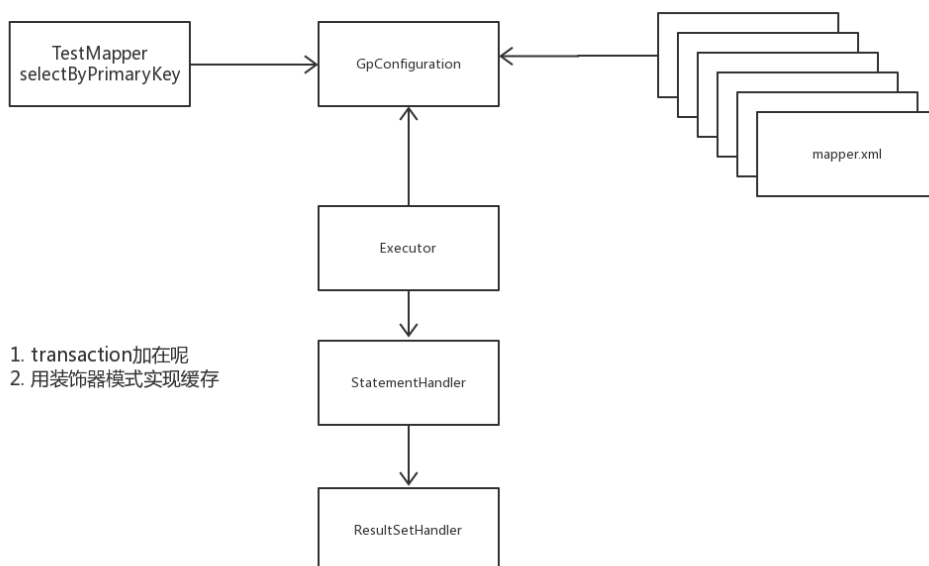


mapping

接口  
sql



1. 配置是写死的
2. 我们的jdbc映射也是写死的
3. Executor职责要单一

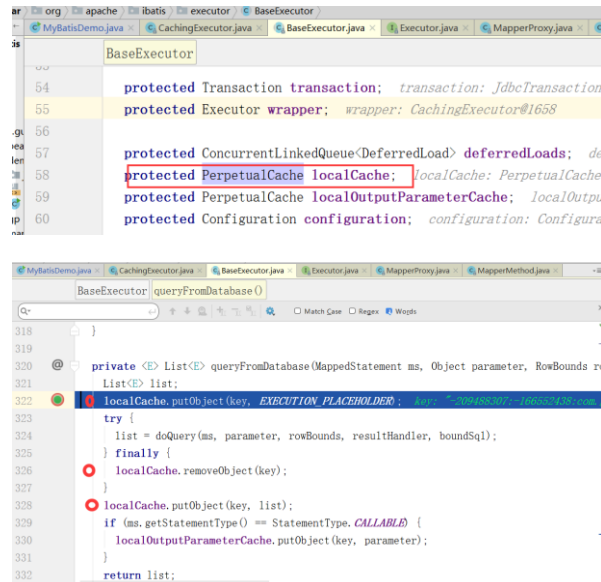


1. transaction加在呢
2. 用装饰器模式实现缓存

## 4.1 缓存

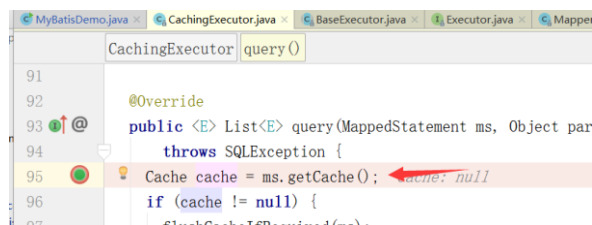
### a. MyBatis 缓存体系

#### 一级缓存

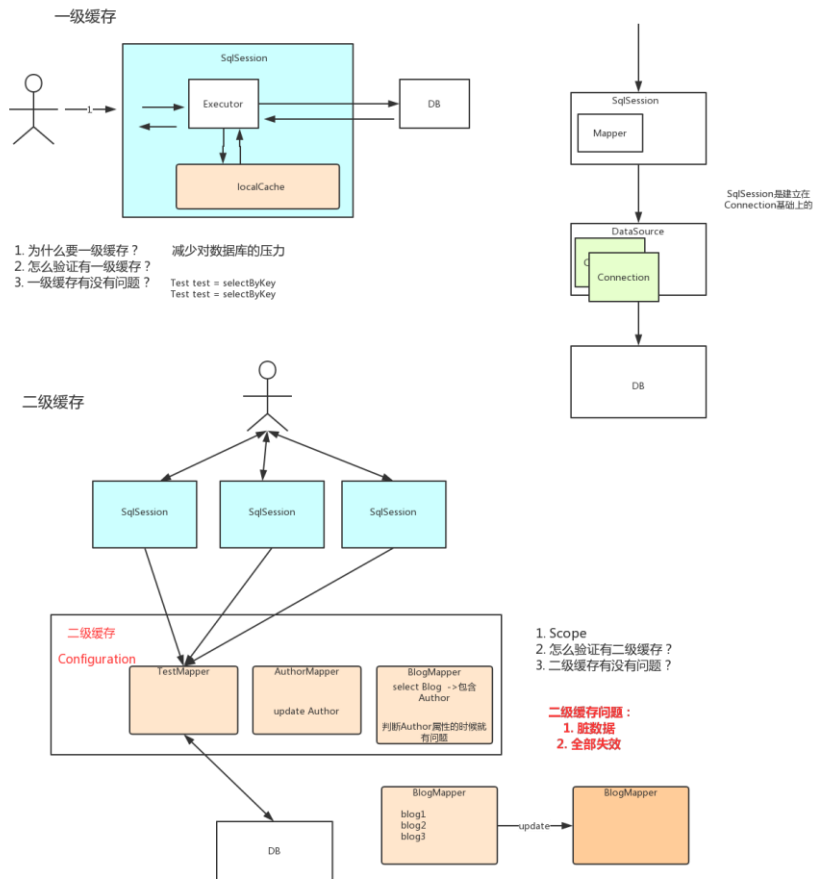


#### 二级缓存

Cache cache = ms.getCache();

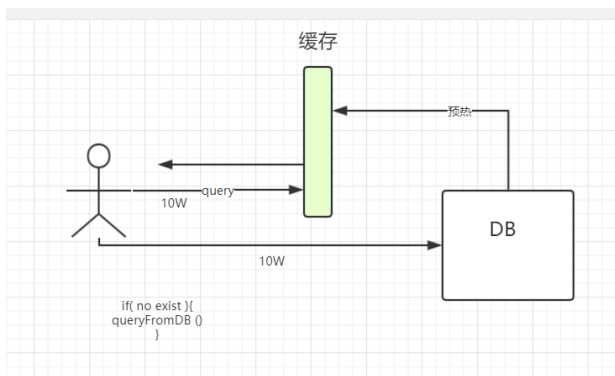






## b. 加入缓存

题外话：缓存击穿



## 4.2 事务

# 5. mybatis-spring-x.x.x.jar

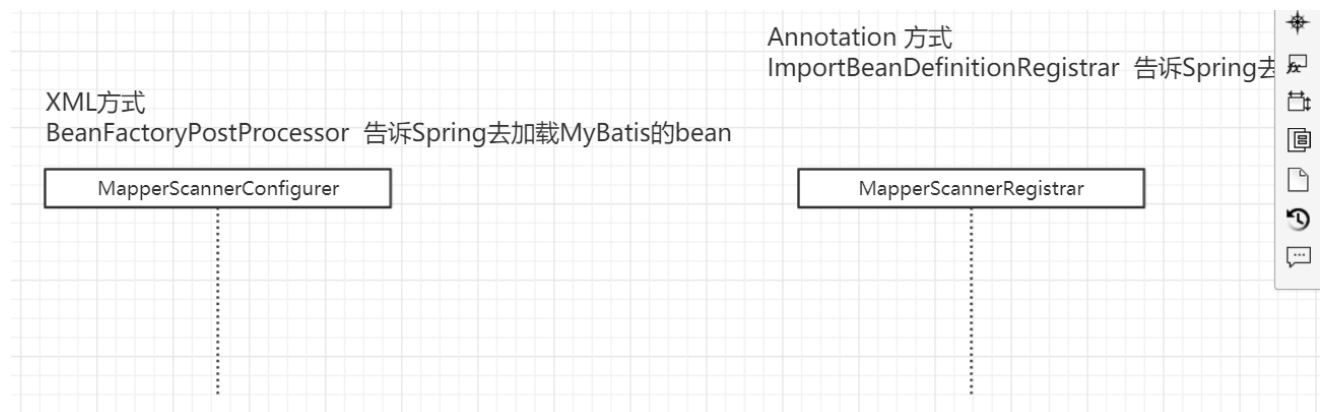
## 5.1 SqlSession\Mapper 生命周期管理

SqlSession 的生命周期在没有 Transaction 的情况下是 request/method 级别的;

在有 Transaction 的情况下, 是 Transaction 范围内的。

Mapper 在 spring 管理时, 是一个全局的 singleton.

1. mapper 是以注入主进来的?



1. mapper 为什么可以是 singleton?

```
186 // but, the actual class of the bean is MapperFactoryBean
187 definition.getConstructorArgumentValues().addGenericArgumentValue(
188     definition.setBeanClass(this.mapperFactoryBean.getClass());
189
```

188 行

```
13 */
14 @Override
15 public T getObject() throws Exception {
16     return getSqlSession().getMapper(this.mapperInterface);
17 }
18
```

5.2 事务管理

## 6. MyBatis VS Hibernate

---

## 7. 回顾总结



做技术人的指路明灯，职场生涯的精神导师。