

# Mybatis源码分析

应用专题之mybatis  
2018年图灵学院VIP二期  
悟空老师(QQ:245553999)

## mybatis 核心概念

Configuration、SqlSessionFactory、Session、Executor、MappedStatement、StatementHandler、ResultSetHandler

名称	意义
Configuration	管理 mysql-config.xml 全局配置关系类
SqlSessionFactory	Session 管理工厂接口
Session	SqlSession 是一个面向用户（程序员）的接口。SqlSession 中提供了很多操作数据库的方法
Executor	执行器是一个接口（基本执行器、缓存执行器） 作用：SqlSession 内部通过执行器操作数据库
MappedStatement	底层封装对象 作用：对操作数据库存储封装，包括 sql 语句、输入输出参数
StatementHandler	具体操作数据库相关的 handler 接口
ResultSetHandler	具体操作数据库返回结果的 handler 接口

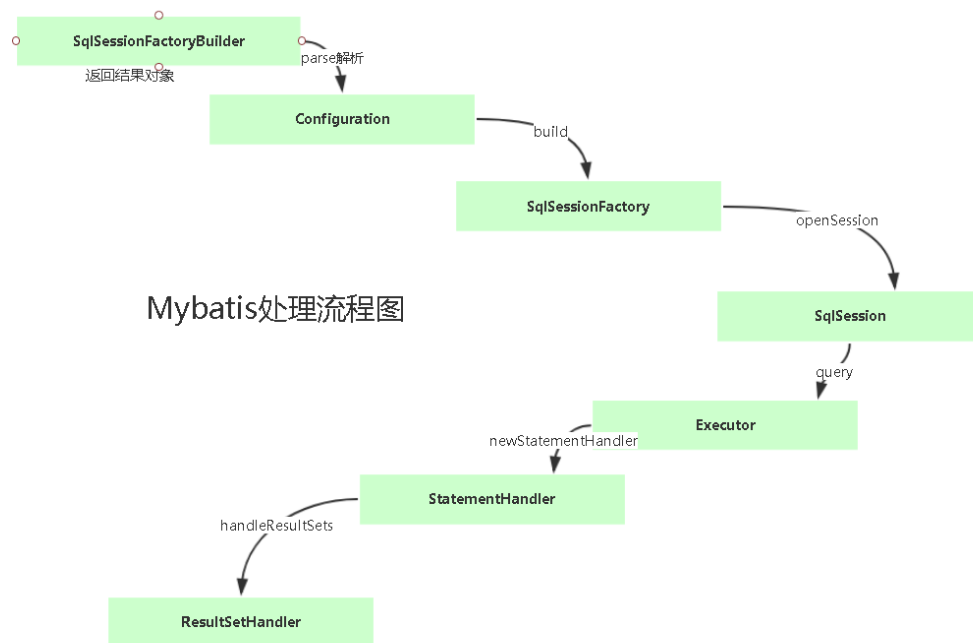
# 整体认识 mybatis 源码包

└─annotations	->注解相关 比如 select insert
└─binding	-> mapper 相关
└─builder	->解析 xml 相关
└─cache	->缓存
└─cursor	-> 返回结果 resultset
└─datasourcer	->数据管理
└─exceptionsr	-> 异常
└─executorr	-> 执行器
└─io	->classloader
└─jdbc	->jdbc
└─lang	->jdk7 jdk8
└─logging	->日志相关
└─mapping	->mapper 相关的封装
└─parsing	->xml 相关解析
└─plugin	->拦截器
└─reflection	->反射相关
└─scripting	->数据厂家
└─session	->session
└─transaction	->事务
└─type	->返回类型对应

# Debug 一行行看代码



宏观 微观 画图



Mybatis处理流程图

## Mybatis 之 Configuration

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

### Properties 文件解析:

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

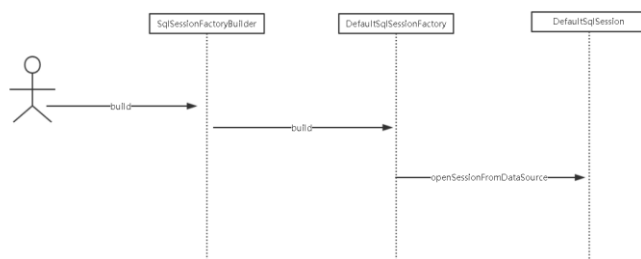
### Setting 文件解析:

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

environments 文件解析:

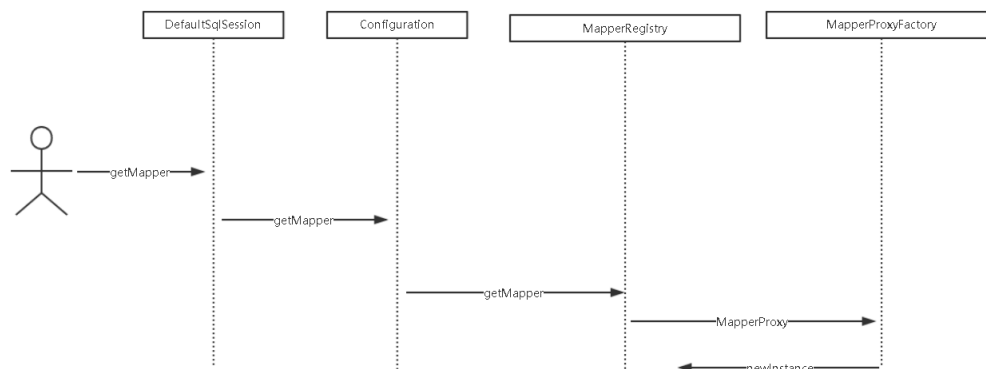
org.apache.ibatis.builder.xml.XMLConfigBuilder#environmentsElement

## Mybatis 之 Session



## Mybatis 之 Mapper

:Type interface com.jiagouedu.UserMapper is not known to the MapperRegistry.

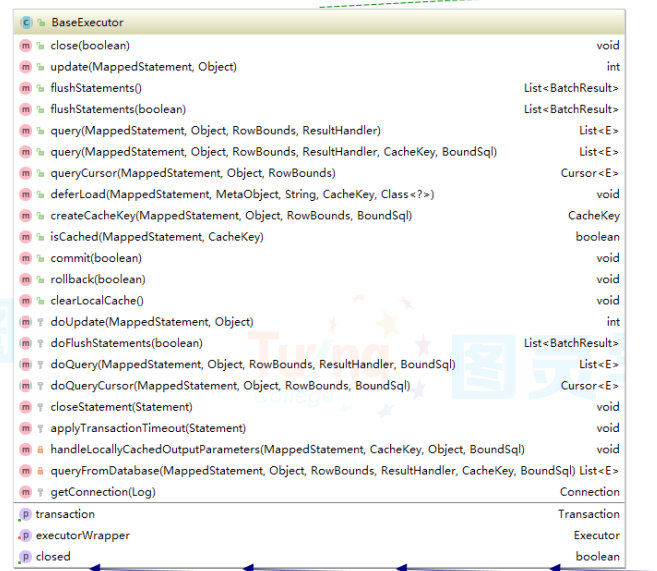
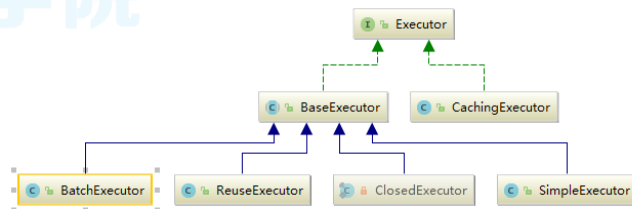


# Mybatis 之 Sql

见 txt.txt 文件

# Mybatis 之 Executor

学院



其实是不干事情的

org.apache.ibatis.executor.statement.StatementHandler

org.apache.ibatis.executor.statement.PreparedStatementHandler

org.apache.ibatis.executor.resultset.ResultSetHandler

org.apache.ibatis.executor.resultset.DefaultResultSetHandler

# Mybatis 之 cache:

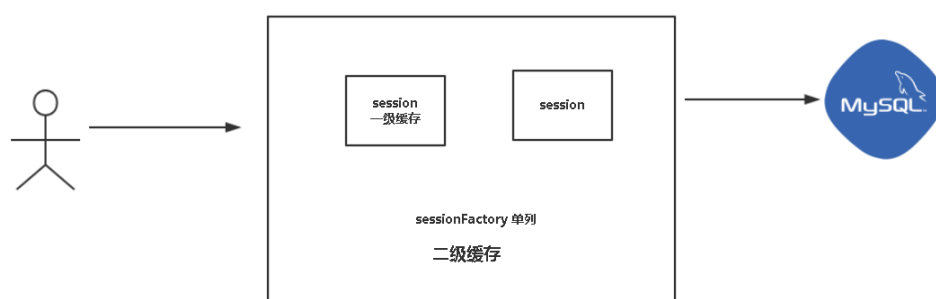
每当我们使用 MyBatis 开启一次和数据库的会话，MyBatis 会创建出一个 SqlSession 对象表示一次数据库会话。在对数据库的一次会话中，我们有可能会反复地执行完全相同的查询语句，如果不采取一些措施的话，每一次查询都会查询一次数据库，而我们在极短的时间内做了完全相同的查询，那么它们的结果极有可能完全相同，由于查询一次数据库的代价很大，这有可能造成很大的资源浪费。

为了解决这一问题，减少资源的浪费，MyBatis 会在表示会话的 SqlSession 对象中建立一个简单的缓存，将每次查询到的结果缓存起来，当下次查询的时候，如果判断先前有个完全一样的查询，会直接从缓存中直接将结果取出，返回给用户，不需要再进行一次数据库查询了。

cache key: id + sql + limit + offset xxx

作用域：一级缓存 session

全局：二级缓存



今天源码分析涉及到的模式是（学员整理）：

1. sqlSessionSessionFactory 工厂
2. build 建造者
3. getInstance , Cache 单例
4. 委派 （单词忘了）装饰
5. InterceptorChain 责任链
- 6 Proxy 代理
7. Execute Command 命令
8. doQuery 模板

# Mybatis 作业

annotations 中的@select 源码加载执行

```
import ...
@Mapper
public interface UserMapper {
    @Results({
        @Result(property = "wukong", column = "desc")
    })
    @Select("select id,username,age,phone,`desc` from user where id=#{id}")
    public User selectUser(Integer id);
}
```

提交地址:

<http://git.jiagouedu.com/java-vip/tuling-mybatis/issues/new>

要求: 流程图+找出代码调用流程