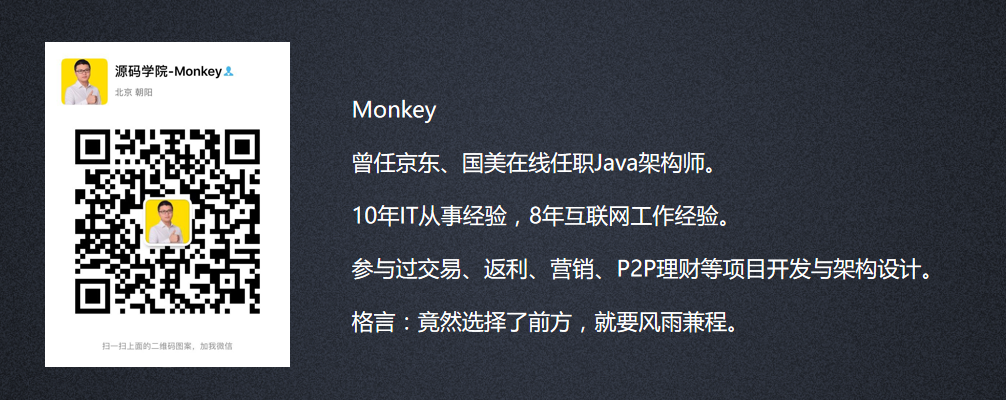
Hi！同学们大家好，我是Monkey老师，首先非常感谢大家对源码学院的支持。

我觉得大家是爱学习的好程序员，白天上班工作，晚上还来学习，别人在玩的时候，你们选择学习，你们放弃了舒适区，选择了挑战自我，选择了学习，其实也是选择了未来。为你们感到骄傲。

这次和大家分享的源码是mybatis，这次会讲倒mybatis的运行原理和缓存机制以及设计模式，站在作者角度思考，徒手实现自己的mybatis。

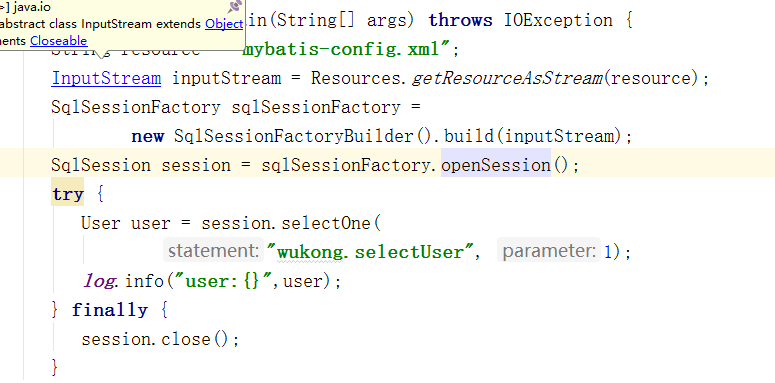


# 传统JDBC的弊端：

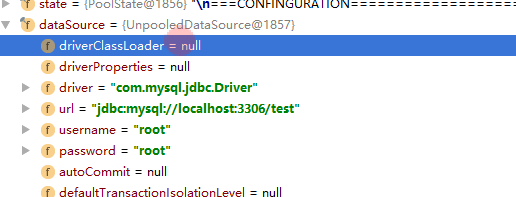
# Mybatis介绍：

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

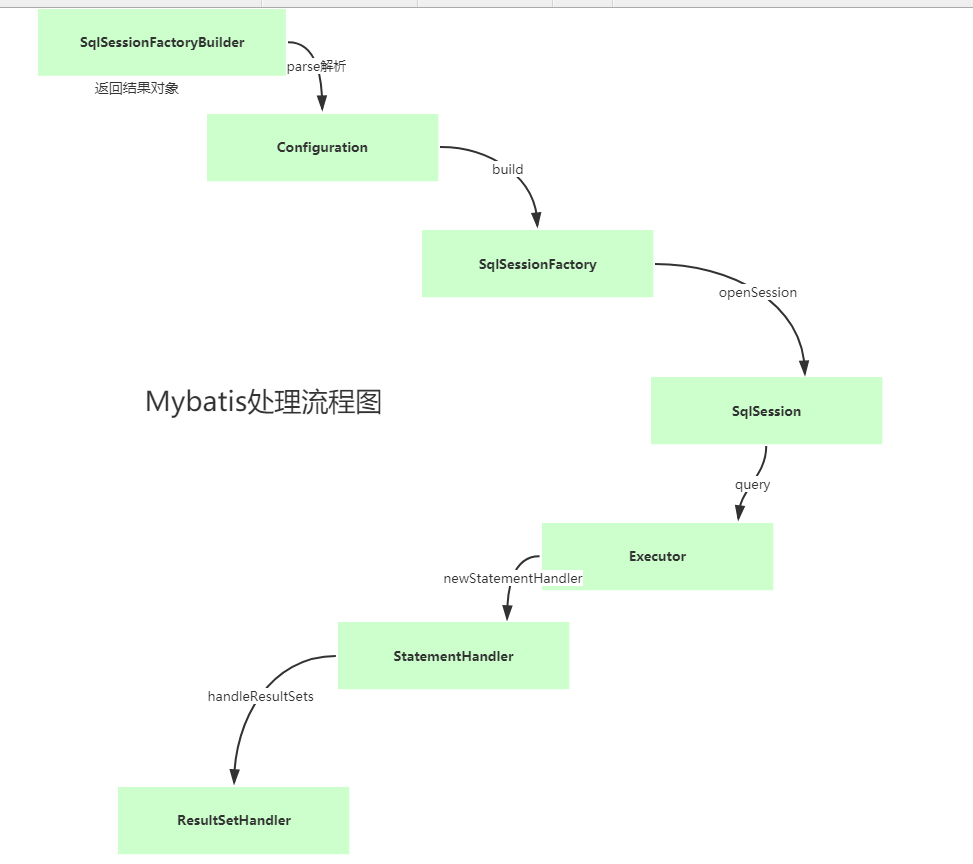
# 源码分析：



一行行的debug看代码了

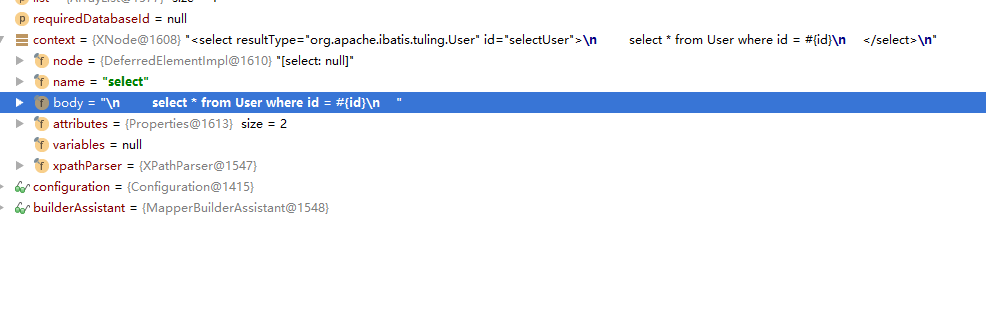


Xml>DataSource



Mybatis加载mapper文件有几种方式？京东面试题 4种

Mybatis 优先级哪个？

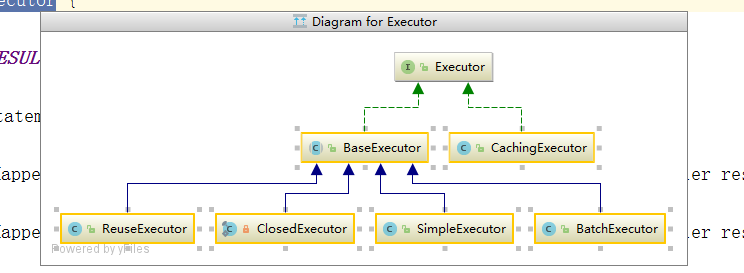


**public enum** ExecutorType {  
 ***SIMPLE***, ***REUSE***, ***BATCH***}

三种执行器

失效

二级缓存 bug 坑



HashMap Map

Mybatis一级缓存默认是开启还是关闭的？ 开启的

# 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 |



# 设计模式：

|  |  |
| --- | --- |
| **Mybatis** | **名称** |
| SqlSessionFactory | 工厂模式 |
| build | 构建模式 |
| GetInstance、Cache | 单列模式 |
| InterceptorChain | 责任链模式 |
| Proxy | 代理模式 |
| Execute | 命令模式 |
| doQuery | 模板模式 |
| Executor|Cache | 装饰模式 |

# 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 ->sessiomn

├─transaction ->事务

└─type ->返回类型对应