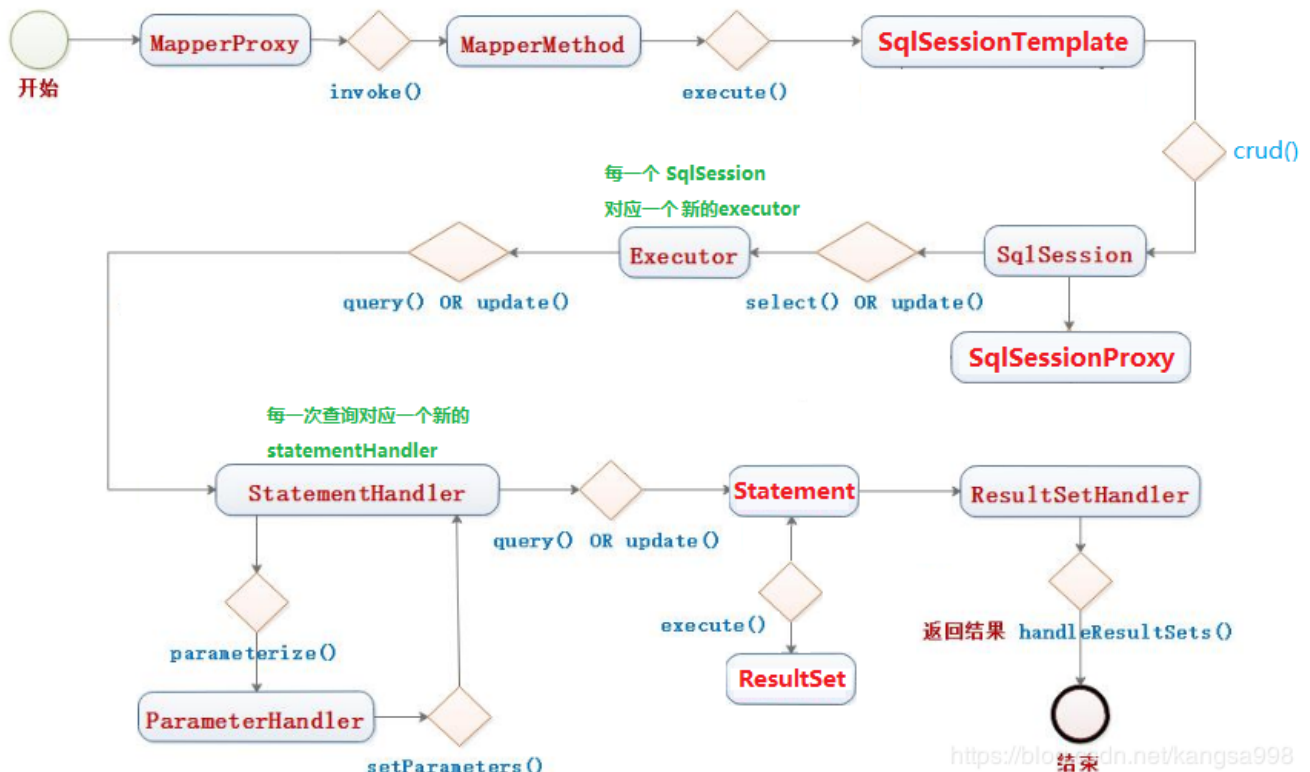


- 1 Mybatis 运行流程图
- 2 源码分析

1 Mybatis 运行流程图



- **MapperProxy**：对 mapper 接口进行实例化的 `InvocationHandler` 类，当调用 mapper 接口时，会调用 `invoke()` 方法，进行处理。MapperProxy 通过 `method` 属性得到 **MapperMethod** 实例
- **MapperMethod**：
 - 将 mapper 接口中的方法参数转化为 `key,value` 形式，以便后续占位符映射
 - 将不同类型的 sql(select/update/delete 等)路由到不同的 **sqlSessionTemplate** 相应的执行方法中 (sqlSessionTemplate .select*/update/delete 等)
- **SqlSessionTemplate**：SqlSessionTemplate 包装了 **SqlSessionFactory** 和 **datasource**，通过它可以创建 **SqlSession** 用于数据库会话
- **SqlSessionProxy**：对 **sqlSessionTemplate** 的拦截，通过 **sqlSessionTemplate** 得到一个 **sqlSession**，并且通过它增加了事务管理功能
- **SqlSession**：在 **SqlSessionProxy** 中获取了 **sqlSession**，其中包含了新建的 **executor**，并且如果事务管理器，此 **sqlSession** 就已支持事务
- **executor**：每创建一次 **sqlSession**，就会创建一个**新的** **executor**，通过不同的 **MappedStatement.executorType** 实例化相应的 **executor**(默认为 **SimpleExecutor**)，同时会将所有相关的 **interceptor** 织入到 **executor** 中，起到扩展作用。默认 **executor** 被 **cacheExecutor** 包装，起到缓存作用，建议关闭(设置 `Configuration.cacheEnabled = false`)
- **interceptor**：创建 **executor** 时，会将相关的 **interceptor** 织入到 **executor** 中，达到增强的目的。即 mybatis 插件功能。需要注意的有

- interceptor 的执行顺序和插入顺序相反（层层包装，先进去的当然最后才被调用）
- 一般情况下通过 invocation.proceed() 方法进行向外层调用，直到执行真正的 executor 方法！如果在 interceptor 中没有调用 invocation.proceed() 方法，就要注意其他 interceptor 是否会失效(没有执行起作用)
- **StatementHandler**：当真正执行 executor 方法时，会**新建一个** StatementHandler。这时又会通过 MappedStatement 中的 statementType 参数，来路由到相应的 StatementHandler 中(一般为 PreparedStatementHandler),并且又通过 interceptor 对 statementHandler 做了织入。创建 statementHandler 的时候，会去**新建** ParameterHandler、ResultSetHandler（创建它们后，又通过 interceptor 对它们进行了织入）
- **Statement**：创建完 statementHandler 就会通过它来创建 statement。真正执行 statementHandler 的数据库操作方法时，会调用 statement 的 execute() 方法来进行数据库操作！！最终调用的是 JdbcPreparedStatement#execute 方法进行的数据库操作
- **KeyGenerator**：当数据库操作完了之后，会调用 mappedStatement#getKeyGenerator 方法，进行主键的更新(如果有且设置的话，默认false)
- **ParameterHandler**：在初始化完 statement 后，会通过 ParameterHandler#setParameters 将参数设置到 statement 中
- **ResultSetHandler**：当执行完 statement 的数据库查询后，会调用 ResultSetHandler#handleResultSets 将结果集映射成 List 形式，并返回
- **TypeHandler**：ParameterHandler、ResultSetHandler 中会通过 TypeHandler 进行参数转化映射

2 源码分析

- 具体源码流程可参考 [github 源码地址](#)
- 具体 mybatis 属性介绍及基础功能可参考 [mybatis 中文文档](#)
- 在整个流程中 **MappedStatement** 这个隐形的类起来很关键的作用，通过它能够得到要执行的 sql 及所有相关信息，可参考 [mappedStatement 解析](#) 作了解