**超详细的 Mybatis 插件开发指南！**

<https://mp.weixin.qq.com/s/NZxerWrXD3NTy_38e5uhKw>

Mybatis插件典型适用场景

**分页功能**

mybatis的分页默认是基于内存分页的（查出所有，再截取），数据量大的情况下效率较低，不过使用mybatis插件可以改变该行为，只需要拦截StatementHandler类的prepare方法，改变要执行的SQL语句为分页语句即可；

**公共字段统一赋值**

一般业务系统都会有创建者，创建时间，修改者，修改时间四个字段，对于这四个字段的赋值，实际上可以在DAO层统一拦截处理，可以用mybatis插件拦截Executor类的update方法，对相关参数进行统一赋值即可；

**性能监控**

对于SQL语句执行的性能监控，可以通过拦截Executor类的update, query等方法，用日志记录每个方法执行的时间；

**其它**

其实mybatis扩展性还是很强的，基于插件机制，基本上可以控制SQL执行的各个阶段，如执行阶段，参数处理阶段，语法构建阶段，结果集处理阶段，具体可以根据项目业务来实现对应业务逻辑。

### Mybatis插件介绍

#### 什么是Mybatis插件

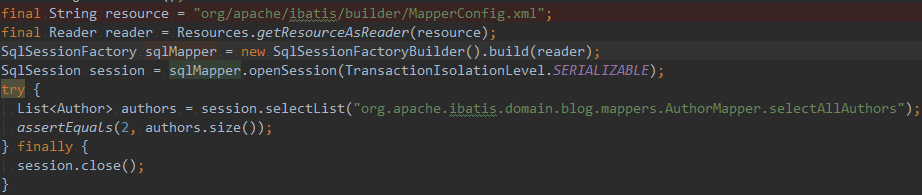
与其称为Mybatis插件，不如叫Mybatis拦截器，更加符合其功能定位，实际上它就是一个拦截器，应用代理模式，在方法级别上进行拦截。

#### 支持拦截的方法

* 执行器Executor（update、query、commit、rollback等方法）；
* 参数处理器ParameterHandler（getParameterObject、setParameters方法）；
* 结果集处理器ResultSetHandler（handleResultSets、handleOutputParameters等方法）；
* SQL语法构建器StatementHandler（prepare、parameterize、batch、update、query等方法）；

#### 拦截阶段

那么这些类上的方法都是在什么阶段被拦截的呢？为理解这个问题，我们先看段简单的代码（摘自mybatis源码中的单元测试SqlSessionTest类），来了解下典型的mybatis执行流程，如下代码所示：



以上代码主要完成以下功能：

* 读取mybatis的xml配置文件信息
* 通过SqlSessionFactoryBuilder创建SqlSessionFactory对象
* 通过SqlSessionFactory获取SqlSession对象
* 执行SqlSession对象的selectList方法，查询结果
* 关闭SqlSession

如下是时序图，在整个时序图中，涉及到mybatis插件部分已标红，基本上就是体现在上文中提到的四个类上，对这些类上的方法进行拦截。

