# 1.概述

MyBatis 允许你在已映射语句执行过程中的某一点进行拦截调用。默认情况下，MyBatis 允许使用插件来拦截的方法调用包括：

* Executor (update, query, flushStatements, commit, rollback, getTransaction, close, isClosed)
* ParameterHandler (getParameterObject, setParameters)
* ResultSetHandler (handleResultSets, handleOutputParameters)
* StatementHandler (prepare, parameterize, batch, update, query)

拦截参数的处理(ParameterHandler)

拦截Sql语法构建的处理(StatementHandler)

拦截执行器的方法(Executor)

拦截结果集的处理(ResultSetHandler)

# 2.1自定义插件1----把Mybatis所有执行的sql都记录下来。

代码实现：

通过对 MyBatis org.apache.ibatis.executor.statement.StatementHandler 中的prepare 方法进行拦截即可。

prepare 方法签名如下：

Statement prepare(Connection connection, Integer transactionTimeout)

throws SQLException;

package extend.use.plugins.customPlugin01;  
  
  
import org.apache.ibatis.executor.statement.StatementHandler;  
import org.apache.ibatis.mapping.BoundSql;  
import org.apache.ibatis.plugin.\*;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
  
import java.sql.Connection;  
import java.util.Properties;  
  
*/\*\*  
 \** ***@Author*** *了个  
 \** ***@date*** *2020/1/8 16:24  
 \*/*@Intercepts({ @Signature(type = StatementHandler.class, method = "prepare", args = { Connection.class, Integer.class}) })  
public class SQLStatsInterceptor implements Interceptor {  
 private final Logger logger = LoggerFactory.*getLogger*(this.getClass());  
  
 @Override  
 public Object intercept(Invocation invocation) throws Throwable {  
  
 StatementHandler statementHandler = (StatementHandler) invocation.getTarget();  
 BoundSql boundSql = statementHandler.getBoundSql();  
 String sql = boundSql.getSql();  
 System.*out*.printf("mybatis intercept sql:%s \n",sql);  
  
 logger.info("mybatis intercept sql:{}", sql);  
 return invocation.proceed();  
 }  
  
 @Override  
 public Object plugin(Object target) {  
 return Plugin.*wrap*(target, this);  
 }  
  
 @Override  
 public void setProperties(Properties properties) {  
 String dialect = properties.getProperty("dialect");  
 logger.info("mybatis intercept dialect:{}", dialect);  
 System.*out*.println(dialect);  
 }  
}

mybatis-config文件中进行如下配置：

<plugins>  
 <plugin interceptor="extend.use.plugins.customPlugin01.SQLStatsInterceptor">  
 <property name="dialect" value="mysql" />  
 </plugin>  
</plugins>

# 2.2 自定义插件----mybatis支持分表

<https://gitee.com/xujian_jason/mybatis-subtable/tree/master/src>

# 3.MyBatis 插件进行分库分表操作

mybatis 插件（plugins） 进行分库分表