Atitit mybatis拦截器机制

<plugins>

<plugin interceptor="attilax.mybatis.sqlInterceptor"/>

</plugins>

package attilax.mybatis;  
  
// attilax.mybatis.sqlInterceptor  
  
  
import java.util.Properties;  
  
import org.apache.ibatis.cache.CacheKey;  
import org.apache.ibatis.executor.Executor;  
import org.apache.ibatis.executor.resultset.ResultSetHandler;  
import org.apache.ibatis.executor.statement.StatementHandler;  
import org.apache.ibatis.mapping.BoundSql;  
import org.apache.ibatis.mapping.MappedStatement;  
import org.apache.ibatis.plugin.\*;  
import org.apache.ibatis.reflection.MetaObject;  
import org.apache.ibatis.reflection.SystemMetaObject;  
import org.apache.ibatis.session.ResultHandler;  
import org.apache.ibatis.session.RowBounds;  
  
import java.sql.\*;  
  
  
@Intercepts({  
 @Signature(type = StatementHandler.class, method = "prepare", args = {Connection.class,Integer.class }),  
 @Signature(type = ResultSetHandler.class, method = "handleResultSets", args = {Statement.class}),  
 @Signature(type = Executor.class, method = "query", args = {MappedStatement.class, Object.class, RowBounds.class, ResultHandler.class, CacheKey.class, BoundSql.class})  
  
  
})  
public class sqlInterceptor implements Interceptor {  
  
 private static final String *SELECT\_ID* = "selectpage";  
  
  
 //插件运行的代码，它将代替原有的方法  
  
 // 该方法写入自己的逻辑  
 @Override  
 public Object intercept(Invocation invocation) throws Throwable {  
 System.*out*.println("PageInterceptor -- intercept");  
  
  
 if (invocation.getTarget() instanceof StatementHandler) {  
 StatementHandler statementHandler = (StatementHandler) invocation.getTarget();  
 BoundSql BoundSql1= statementHandler.getBoundSql();  
 String sql = BoundSql1.getSql();  
  
 System.*out*.println(sql);  
  
 MetaObject MetaObject1 = SystemMetaObject.*forObject*(statementHandler);  
 MappedStatement mappedStatement = (MappedStatement) MetaObject1.getValue("delegate.mappedStatement");  
 String selectId = mappedStatement.getId();  
  
  
//  
//  
// // metaStatementHandler.setValue("delegate.boundSql.sql", pageSql);  
//  
  
  
 }  
  
 return invocation.proceed();  
 }  
  
 */\*\*  
 \* this methd invoke first ,,then intercepte method  
 \* 拦截类型StatementHandler  
 \* // 调用插件  
 \* return Plugin.wrap(target, this);  
 \*/* @Override  
 public Object plugin(Object target) {  
  
 // 调用插件  
 // return Plugin.wrap(target, this);  
 System.*out*.println(" plugin()");  
 if (target instanceof StatementHandler) {  
 return Plugin.*wrap*(target, this);  
 } else {  
 return target;  
 }  
 }  
  
 @Override  
 public void setProperties(Properties properties) {  
  
 }  
  
  
}