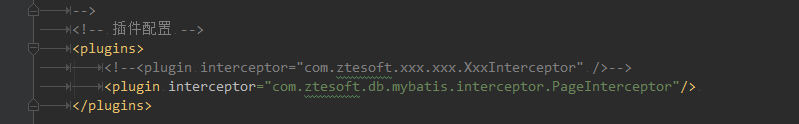
Mybatis 分页

配置文集 配置拦截器



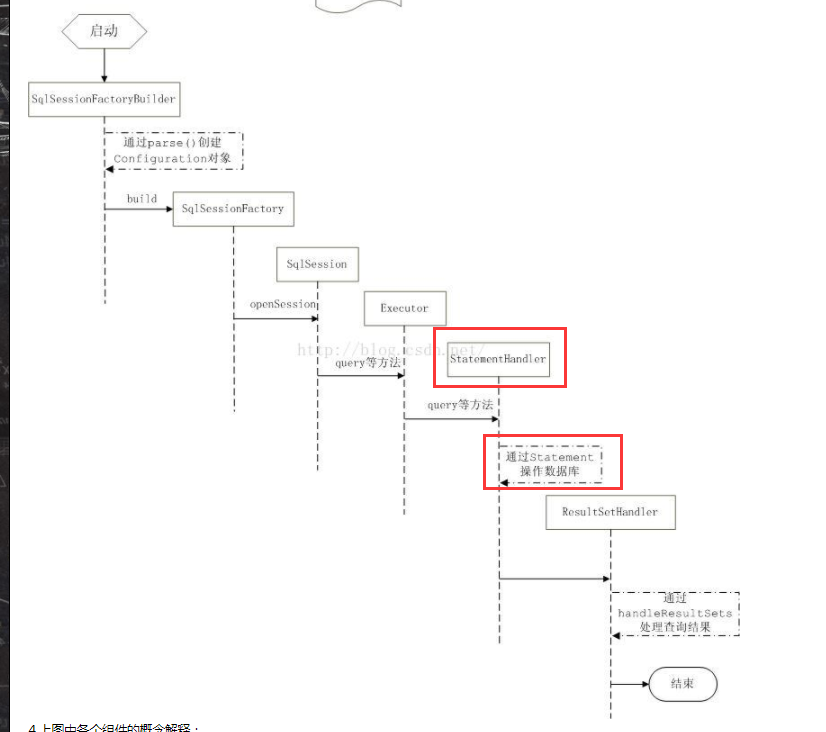
原理参考博文：

( http://blog.csdn.net/ABCD898989/article/details/51244351)

(<http://blog.csdn.net/qi49125/article/details/52277679>)

Mybatis 架构

通过拦截<code>StatementHandler</code>的<code>prepare</code>方法，重写sql语句实现物理分页。



SqlSessionFactoryBuilder：每一个Mybatis的应用程序的入口是SqlSessionFactoryBuilder，它的作用是通过XML配置文件创建Configuration对象（当然也可以在程序中自行创建），然后通过build方法创建SqlSessionFactory对象。没有必要每次访问Mybatis就创建一次SqlSessionFactoryBuilder，通常的做法是创建一个全局的对象就可以了

SqlSessionFactory：SqlSessionFactory对象由SqlSessionFactoryBuilder创建。它的主要功能是创建SqlSession对象(全局变量)

SqlSession：SqlSession对象的主要功能是完成一次数据库的访问和结果的映射，它类似于数据库的session概念，由于\*\*不是线程安全\*\*的，所以SqlSession对象的\*\*作用域需限制在方法内\*\*。SqlSession的默认实现类是DefaultSqlSession，它有两个必须配置的属性：Configuration和Executor。

Configuration即是我们的mybatis-config.xml文件, SqlSession对数据库的操作都是通过Executor来完成的

SqlSession的getMapper方法是联系应用程序和Mybatis纽带，应用程序访问getMapper时，Mybatis会根据传入的接口类型和对应的XML配置文件生成一个代理对象，这个代理对象就叫Mapper对象。\*\*应用程序获得Mapper对象后，就应该通过这个Mapper对象来访问Mybatis的SqlSession对象\*\*，这样就达到里插入到Mybatis流程的目的。

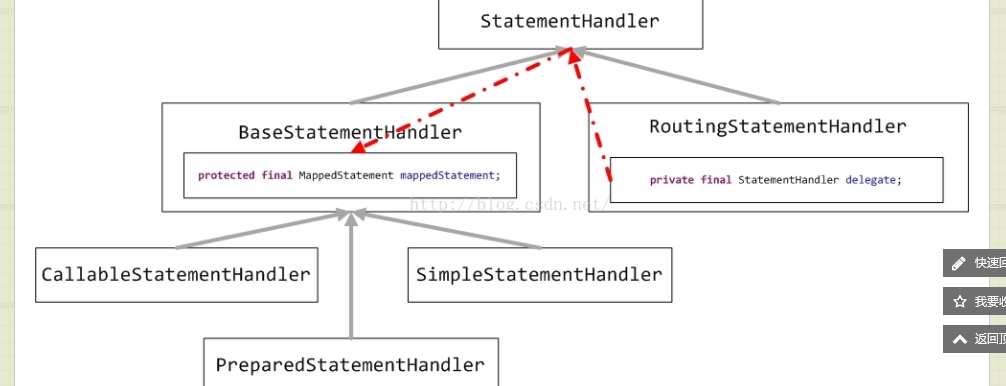
拦截器的对应注解内容如下：【继承自Mybatis的Interceptor】

@Intercepts({@Signature(type=StatementHandler.class,method="prepare",args={Connection.class})})

已经成功的拦截到了目标对象，然后，就开始要改变查询过程了。在这里，出现了两个基本的问题

如何获取原始的查询语句，即mapper文件中的sql语句。

如何获取分页信息。



几个重要的类

StatementHandler 需要拦截类，用于获取所有的配置信息。是一个接口。

MetaObject： 上面的类的实例化的类

方法如下:

MetaObject metaStatementHandler = MetaObject.forObject(statementHandler, DEFAULT\_OBJECT\_FACTORY,

DEFAULT\_OBJECT\_WRAPPER\_FACTORY, DEFAULT\_OBJECT\_REFLECTOR\_FACTORY);

Configuration 全局配置类，可以从MetaObject获取

MappedStatement 包含所有要找的sql信息，可以从MetaObject获取

BoundSql：用于获取sql 语句

重新设置sql语句

metaStatementHandler.setValue("delegate.boundSql.sql", pageSql);

获取jdbc connection

Connection connection = (Connection) invocation.getArgs()[0];

统计所有个数

1. Connection connection = (Connection) invocation.getArgs()[0];
2. //利用原始sql语句的方法执行
3. PreparedStatement countStatement = connection.prepareStatement(countSql);
4. //在本例中，查询参数为空，但实际应用时，多为带有条件的分页查询，下面的这句话就是获取查询条件的参数
5. ParameterHandler parameterHandler = (ParameterHandler) metaObject.getValue("delegate.parameterHandler");
6. //经过set方法，就可以正确的执行sql语句
7. parameterHandler.setParameters(countStatement);
8. ResultSet rs = countStatement.executeQuery();