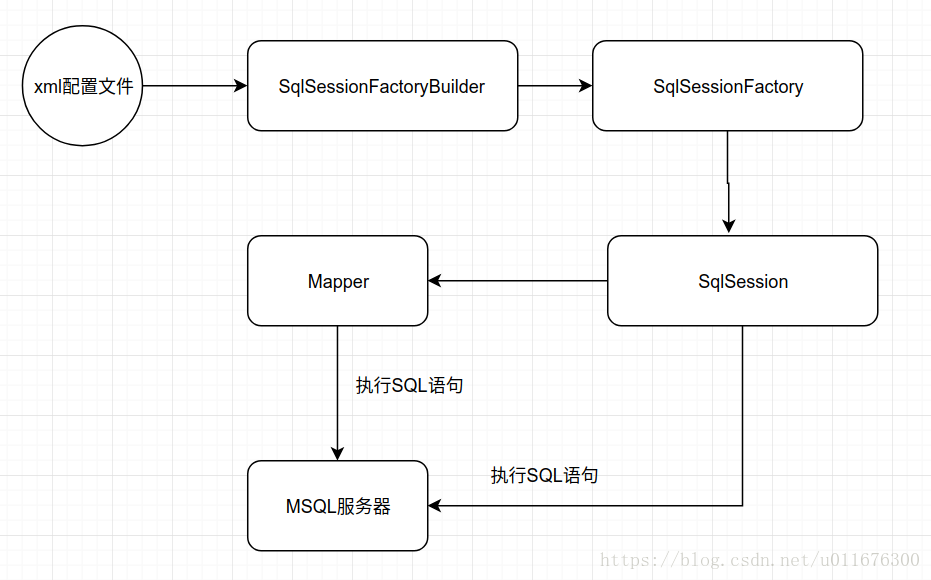
**MyBatis主要包含的核心部件**

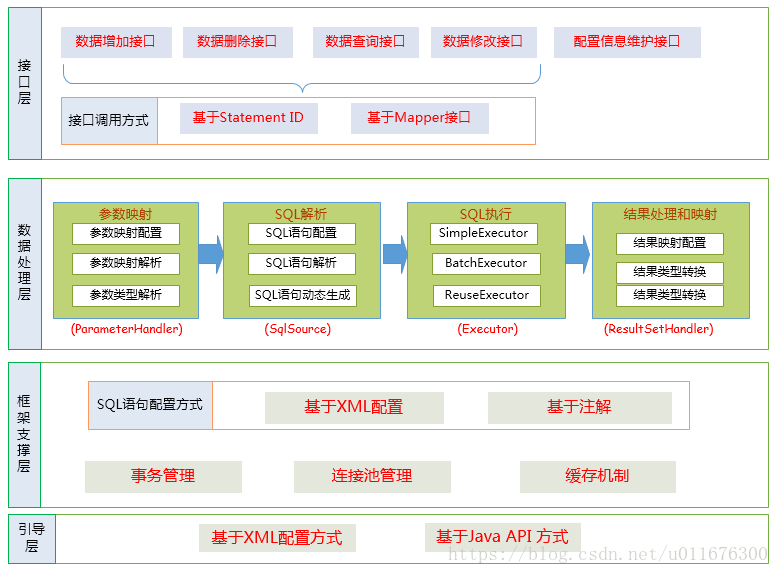
* SqlSession : Mybatis工作的主要顶层API，表示和数据库交互的会话，完成必要数据库增删改查功能。这是MyBatis主要的一个类，用来执行SQL，获取映射器，管理事务。通常情况下，在程序中使用的Mybatis的API就是这个接口定义的方法。
* Executor: Mybatis执行器，是Mybatis调度的核心，负责SQL语句的生成和查询缓存的维护
* StatementHandler: 封装了JDBC Statement操作，负责对JDBC statement的操作，如设置参数、将Statement结果集转换成List集合。
* ParameterHandler :参数处理器，负责对用户传递的参数转换成JDBC Statement所需要的参数
* ResultSetHandler ：结果集处理器，负责将JDBC返回的ResultSet结果集对象转换成List类型的集合
* TypeHandler: 类型处理器，负责java数据类型和jdbc数据类型之间的映射和转换
* MappedStatement :映射语句，维护了一条<select|update|delete|insert>节点的封装
* SqlSource: SQL源码，负责根据用户传递的parameterObject,动态地生成SQL语句，将信息封装到BoundSql对象中，并返回
* BoundSql:绑定的SQL，是从SqlSource而来，将动态内容都处理完得到的SQL语句字符串，其中包括?，还有绑定的参数
* Configuration:Mybatis所有的配置信息都维护在Configuration对象之中
* MapperProxyFactory：映射器代理工厂，主要是使用jdk代理mapper的实现以实现调用SqlSessoin,接口类名称对应配置文件的namespace的值，等标签的id对应方法名称。

**MyBatis基本构成**



* xml配置文件，[参考这里](https://blog.csdn.net/u011676300/article/details/82904755)
* SqlSessionFactoryBuilder  
  解析xml配置文件，创建Configration对象，并根据配置创建SqlSessionFactory
* SqlSessionFactory  
  通过SqlSessionFactory创建SqlSession对象
* SqlSession  
  可以通过其发送SQL语句去执行并返回结果，或者获取mapper对象
* Mapper  
  可以通过其发送SQL语句去执行并返回结果

**MyBatis整体架构**



**接口层**

用于发送sql语句去执行并返回结果。该模块的核心类是：

org.apache.ibatis.session.SqlSessionFactoryBuilder;

org.apache.ibatis.session.SqlSessionFactory;

org.apache.ibatis.session.SqlSession;

MyBatis和数据库的交互有两种方式：

* 使用传统的MyBatis提供的API；
* 使用Mapper接口

获取SqlSession

SqlSessionFactory factory;

SqlSession sqlSession;

try {

Reader reader = Resources.getResourceAsReader("mybatis-config.xml");

factory = new SqlSessionFactoryBuilder().build(reader);

sqlSession = factory.openSession();

} catch (Exception ex) {

ex.printStackTrace();

}

* 使用传统的MyBatis提供的API；

List<SysUser> sysUsers = sqlSession.selectList("com.mybatis.dao.mapper.sysUserMapper.selectAll");

* 使用Mapper接口

sysUserMapper = sqlSession.getMapper(SysUserMapper.class);

List<SysUser> sysUsers = sysUserMapper.selectAll();

从上面可知使用传统方式书写起来非常不方便，一般都是使用Mapper方式

通过SqlSession.getMapper(XXXMapper.class) 方法，MyBatis会根据相应的接口声明的方法信息，通过动态代理机制生成一个Mapper实例，我们使用Mapper接口的某一个方法时，MyBatis会根据这个方法的方法名和参数类型，确定Statement Id，底层还是通过SqlSession.select(“statementId”,parameterObject);或者SqlSession.update(“statementId”,parameterObject);等等来实现对数据库的操作

**数据处理层**

数据处理层主要是实现参数映射，拼装SQL语句，将SQL执行的结果映射到JAVA对象。