## MyBatis简介

本是Apache的一个开源项目iBatis，2010年这个项目有apache迁移到了google，

并更名为MyBatis，2013年迁移到GitHub。

基于Java的持久层框架，内部封装了JDBC，使开发者只需关注SQL语句本身。

MyBatis通过xml或注解的方式将要执行的各种statement(statement、preparedStatement等)配置起来。并通过Java对象和Statement中的SQL的动态参数进行映射生成最终执行的SQL语句，最后由MyBatis框架执行SQL语句并将结果映射成**Java对象**并返回。

MyBatis在三层架构中处于DAO层，用于对数据库实现增删改查。

## 

## MyBatis与Hibernate

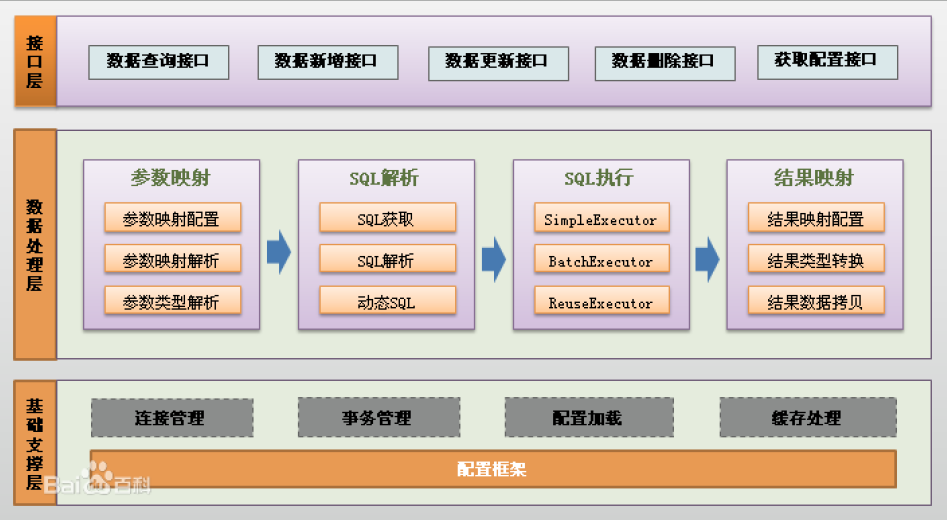
Hibernate框架是提供了全面的数据库封装机制的“全自动”ORM(Object Relational Mapping对象关系映射)，即实现了POJO和数据库表之间的映射，以及SQL的自动生成和执行。

MyBatis只能算“半自动”ORM。着力点在POJO类与SQL语句之间的映射关系。MyBatis不会为程序员自动生成SQL语句，具体SQL语句需要程序员自己编写，然后经过SQL语句映射文件，将SQL所需的参数，以及返回的结果字段映射到指定的POJO。

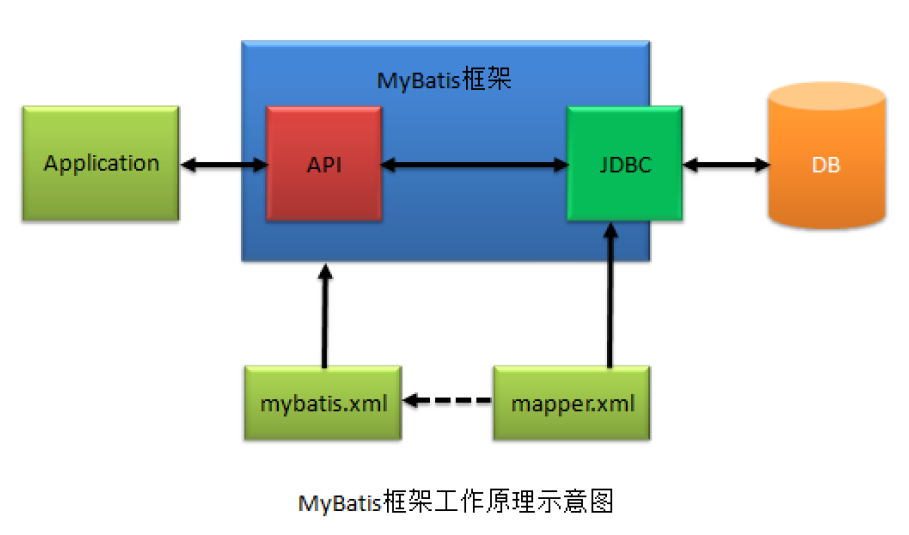
### MyBatis特点:

* 在xml文件中配置SQL语句，实现了SQL语句与代码的分离，方便维护程序。
* 程序员可以结合数据库自身的特点灵活控制SQL语句，实现比Hibernate等全自动ORM框架更高的查询效率，能够完成复杂查询。
* 简单，易于学习，上手快。

## MyBatis体系结构



## MyBatis工作原理



# ORM

# MyBatis工作原理原型图

# 工作原理解析

mybatis应用程序通过***SqlSessionFactoryBuilder***从mybatis-config.xml配置文件（也可以用Java文件配置的方式，需要添加@Configuration）中构建出***SqlSessionFactory***（SqlSessionFactory是线程安全的）；然后，SqlSessionFactory的实例直接开启一个SqlSession，再通过SqlSession实例获得Mapper对象并运行Mapper映射的SQL语句，完成对数据库的CRUD和事务提交，之后关闭SqlSession。

说明：SqlSession是单线程对象，因为它是非线程安全的，是持久化操作的独享对象，类似jdbc中的Connection，底层就封装了jdbc连接。

详细流程如下：

1、加载mybatis全局配置文件（数据源、mapper映射文件等），解析配置文件，MyBatis基于XML配置文件生成Configuration，和一个个MappedStatement（包括了参数映射配置、动态SQL语句、结果映射配置），其对应着<select | update | delete | insert>标签项。  
  
2、SqlSessionFactoryBuilder通过Configuration对象生成SqlSessionFactory，用来开启SqlSession。  
  
3、SqlSession对象完成和数据库的交互：  
a、用户程序调用mybatis接口层api（即Mapper接口中的方法）  
b、SqlSession通过调用api的Statement ID找到对应的MappedStatement对象  
c、通过Executor（负责动态SQL的生成和查询缓存的维护）将MappedStatement对象进行解析，sql参数转化、动态sql拼接，生成jdbc Statement对象  
d、JDBC执行sql。

e、借助MappedStatement中的结果映射关系，将返回结果转化成HashMap、JavaBean等存储结构并返回。