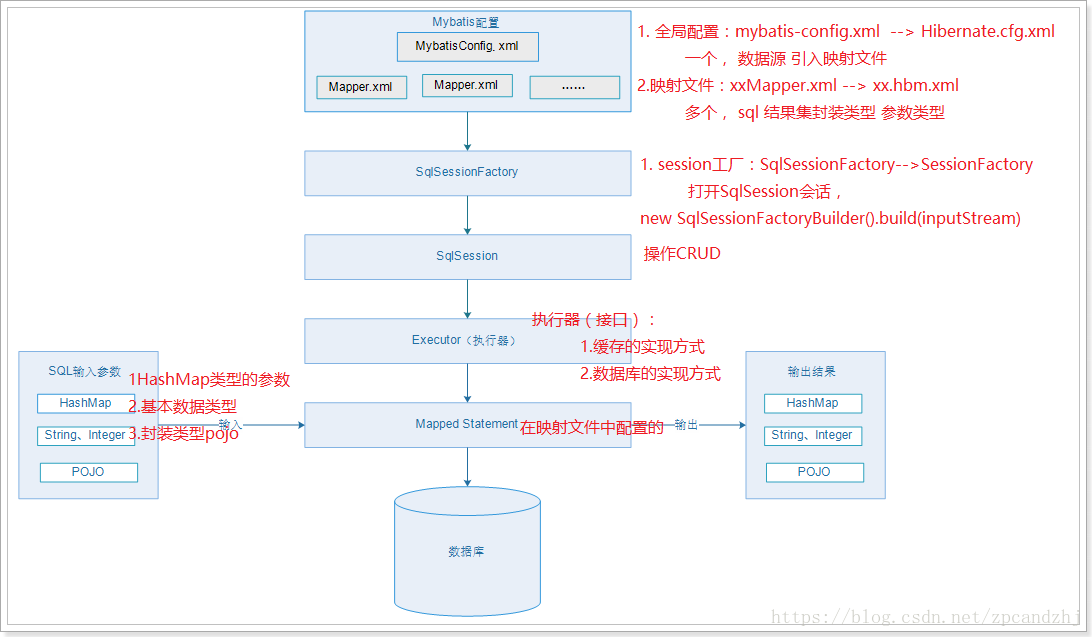
MyBatis

1. 框架介绍

MyBatis是一款优秀的持久层框架，它支持定制化SQL、存储过程以及高级映射。MyBatis避免了几乎所有的 JDBC 代码和手动设置参数以及获取结果集。MyBatis可以使用简单的 XML 或注解来配置和映射原生信息，将接口和Java的POJOs(Plain Ordinary Java Object,普通的 Java对象)映射成数据库中的记录。Mybatis主要做了两点工作：

* 1. 封装JDBC操作。
  2. 利用反射打通Java类与SQL语句之间的相互转换。

1. Mybatis的整体框架



* 1. mybaits-config.xml配置文件：是Mybatis的全局配置文件，包含了影响Mybatis行为的设置和属性信息，该配置文件的层级结构如下所示：
* configuration 配置
  + [properties 属性](mk:@MSITStore:D:\java资料\MyBatis%20中文帮助文档\mybatis.CHM::/configuration.html#properties)
  + [settings 设置](mk:@MSITStore:D:\java资料\MyBatis%20中文帮助文档\mybatis.CHM::/configuration.html#settings)
  + [typeAliases 类型命名](mk:@MSITStore:D:\java资料\MyBatis%20中文帮助文档\mybatis.CHM::/configuration.html#typeAliases)
  + [typeHandlers 类型处理器](mk:@MSITStore:D:\java资料\MyBatis%20中文帮助文档\mybatis.CHM::/configuration.html#typeHandlers)
  + [objectFactory 对象工厂](mk:@MSITStore:D:\java资料\MyBatis%20中文帮助文档\mybatis.CHM::/configuration.html#objectFactory)
  + [plugins 插件](mk:@MSITStore:D:\java资料\MyBatis%20中文帮助文档\mybatis.CHM::/configuration.html#plugins)
  + [environments 环境](mk:@MSITStore:D:\java资料\MyBatis%20中文帮助文档\mybatis.CHM::/configuration.html#environments)
    - environment 环境变量
      * transactionManager 事务管理器
      * dataSource 数据源
  + [databaseIdProvider chinese?](mk:@MSITStore:D:\java资料\MyBatis%20中文帮助文档\mybatis.CHM::/configuration.html#databaseIdProvider)
  + [mappers 映射器](mk:@MSITStore:D:\java资料\MyBatis%20中文帮助文档\mybatis.CHM::/configuration.html#mappers)

具体的XML配置文件如下：

1. **<?xml** version="1.0" encoding="UTF-8" **?>**
2. <!DOCTYPE configuration
3. PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
4. "http://mybatis.org/dtd/mybatis-3-config.dtd"**>**
5. <!-- 配置文件的根元素 -->
6. **<configuration>**
7. <!-- 属性：定义配置外在化 -->
8. **<properties></properties>**
9. <!-- 设置：定义mybatis的一些全局性设置 -->
10. **<settings>**
11. <!-- 具体的参数名和参数值 -->
12. **<setting** name="" value=""**/>**
13. **</settings>**
14. <!-- 类型名称：为一些类定义别名 -->
15. **<typeAliases></typeAliases>**
16. <!-- 类型处理器：定义Java类型与数据库中的数据类型之间的转换关系 -->
17. **<typeHandlers></typeHandlers>**
18. <!-- 对象工厂 -->
19. **<objectFactory** type=""**></objectFactory>**
20. <!-- 插件：mybatis的插件,插件可以修改mybatis的内部运行规则 -->
21. **<plugins>**
22. **<plugin** interceptor=""**></plugin>**
23. **</plugins>**
24. <!-- 环境：配置mybatis的环境 -->
25. **<environments** default=""**>**
26. <!-- 环境变量：可以配置多个环境变量，比如使用多数据源时，就需要配置多个环境变量 -->
27. **<environment** id=""**>**
28. <!-- 事务管理器 -->
29. **<transactionManager** type=""**></transactionManager>**
30. <!-- 数据源 -->
31. **<dataSource** type=""**></dataSource>**
32. **</environment>**
33. **</environments>**
34. <!-- 数据库厂商标识 -->
35. **<databaseIdProvider** type=""**></databaseIdProvider>**
36. <!-- 映射器：指定映射文件或者映射类 -->
37. **<mappers></mappers>**
38. **</configuration>**
    1. XXXMapper.xml映射文件：MyBatis的核心所在，MyBatis 的构建就是聚焦于 SQL语句的,使其远离于普通的JDBC方式，其中主要的属性有：

* cache – 配置给定命名空间的缓存。
* cache-ref – 从其他命名空间引用缓存配置。
* resultMap – 最复杂,也是最有力量的元素,用来描述如何从数据库结果集中来加载你的对象。
* sql – 可以重用的 SQL 块,也可以被其他语句引用。
* insert – 映射插入语句
* update – 映射更新语句
* delete – 映射删除语句
* select – 映射查询语句

1. MyBatis的工作原理

我们把Mybatis的功能架构分为三层：

1. API接口层：提供给外部使用的接口API，开发人员通过这些本地API来操纵数据库 接口层一接收到调用请求就会调用数据处理层来完成具体的数据处理
2. 数据处理层：负责具体的SQL查找、SQL解析、SQL执行和执行结果映射处理等 它主要的目的是根据调用的请求完成一次数据库操作
3. 基础支撑层：负责最基础的功能支撑，包括连接管理、事务管理、配置加载和缓存处理，这些都是共用的东西，将他们抽取出来作为最基础的组件 为上层的数据处理层提供最基础的支撑

