# 概述

## 架构示意图



# 知识点

## 配置文件

配置文件是mybatis的入口，其中包含很多可配置项，各个配置项含义参考<http://www.mybatis.org/mybatis-3/configuration.html>，翻译后的网站

<http://www.mybatis.org/mybatis-3/zh/configuration.html>，对于每个配置项，需要注意的事项包括如下：

### properties

设置Property包括三种方式：

* 通过property文件设置，然后将文件路径赋值给properties的resource属性；
* 通过property属性逐个设置；
* 代码调用mybatis的api时提供的参数；

比如

<properties resource="org/mybatis/example/config.properties">

<property name="username" value="dev\_user"/>

<property name="password" value="F2Fa3!33TYyg"/>

</properties>

注意，如果在编写配置文件时，一个属性不止一次被配置时，mybatis按照如下顺序进行加载：

* 在 properties 元素体内指定的属性首先被读取。
* 然后根据 properties 元素中的 resource 属性读取类路径下属性文件或根据 url 属性指定的路径读取属性文件，并覆盖已读取的同名属性。
* 最后读取作为方法参数传递的属性，并覆盖已读取的同名属性。

最先被加载的项，可能会被后续操作给覆盖，因此，通过方法参数传递的属性具有最高优先级，resource/url 属性中指定的配置文件次之，最低优先级的是 properties 属性中指定的属性。比如：

配置文件中设置了username，但是在代码中创建连接时传递了新的username，最后将以方法的参数值为准。

### settings

这一个标签用途很大，包含丰富的配置项，下面是一个示例，具体的使用参考官网

<settings>

<setting name="cacheEnabled" value="true"/>

<setting name="lazyLoadingEnabled" value="true"/>

<setting name="multipleResultSetsEnabled" value="true"/>

<setting name="useColumnLabel" value="true"/>

<setting name="useGeneratedKeys" value="false"/>

<setting name="autoMappingBehavior" value="PARTIAL"/>

<setting name="autoMappingUnknownColumnBehavior" value="WARNING"/>

<setting name="defaultExecutorType" value="SIMPLE"/>

<setting name="defaultStatementTimeout" value="25"/>

<setting name="defaultFetchSize" value="100"/>

<setting name="safeRowBoundsEnabled" value="false"/>

<setting name="mapUnderscoreToCamelCase" value="false"/>

<setting name="localCacheScope" value="SESSION"/>

<setting name="jdbcTypeForNull" value="OTHER"/>

<setting name="lazyLoadTriggerMethods" value="equals,clone,hashCode,toString"/>

</settings>

### typeAliases

顾名思义，为类型设置别名，举例：

<!-- 取别名 -->

<typeAliases>

<typeAlias type="com.lc.model.User" alias="user"/>

</typeAliases>

## 接口层

* 掌握Mybatis使用的两种方式：
  + 一种是传统API方式，通过传递StatementId和Arguments访问mapper中的sql，参考App中的测试代码；
  + 一种是使用Mapper接口，这种方式不需要实现Mapper接口，但是方法名必须和Mapper XML中StatementId保持一致；
* 使用的基本步骤：
  + 构造Mybatis的构造文件，包括db.driver，username，password等等；
  + 创建SqlSessionFactory
  + 从SqlSessionFactory中获取一条session
  + 如果是Mapper接口方式，通过session.getMapper(mapper接口名)获取一个Mapper，比如UserMapper userMapper = session.getMapper(UserMapper.class);这种方式也可以通过注解的方式表明sql语句
  + 如果是传统API方式+StatementId，直接利用session.select|insert|update|delete（key，parameter）,其中key是Mapper XML文件中的StatementId，比如：userList = session.selectList("userMapper.getUserByParam",params);



Figure 1 API模式



Figure 2 Mapper接口模式

参考代码

MybatisDemo项目

## 数据处理层

## 引导层

# 知识点

# 参考

《深入理解mybatis原理》 MyBatis的架构设计以及实例分析

<https://blog.csdn.net/luanlouis/article/details/40422941>