[**MyBatis学习总结(三)——优化MyBatis配置文件中的配置**](http://www.cnblogs.com/xdp-gacl/p/4264301.html)

**一、连接数据库的配置单独放在一个properties文件中**

　　之前，我们是直接将数据库的连接配置信息写在了MyBatis的conf.xml文件中，如下：

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="UTF-8"?>

2 <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd">

3 <configuration>

4 <environments default="development">

5 <environment id="development">

6 <transactionManager type="JDBC" />

7 <!-- 配置数据库连接信息 -->

8 <dataSource type="POOLED">

9 <property name="driver" value="com.mysql.jdbc.Driver" />

10 <property name="url" value="jdbc:mysql://localhost:3306/mybatis" />

11 <property name="username" value="root" />

12 <property name="password" value="XDP" />

13 </dataSource>

14 </environment>

15 </environments>

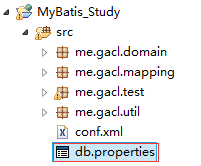
16

17 </configuration>

[复制代码](javascript:void(0);)

　　其实我们完全可以将数据库的连接配置信息写在一个properties文件中，然后在conf.xml文件中引用properties文件，具体做法如下：

　　1、在src目录下新建一个db.properties文件，如下图所示：



　　在db.properties文件编写连接数据库需要使用到的数据库驱动，连接URL地址，用户名，密码，如下：

1 driver=com.mysql.jdbc.Driver

2 url=jdbc:mysql://localhost:3306/mybatis

3 name=root

4 password=XDP

　　2、在MyBatis的conf.xml文件中引用db.properties文件，如下：

[复制代码](javascript:void(0);)

1 <?xml version="1.0" encoding="UTF-8"?>

2 <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd">

3 <configuration>

4

5 <!-- 引用db.properties配置文件 -->

6 <properties resource="db.properties"/>

7 <!--

8 development : 开发模式

9 work : 工作模式

10 -->

11 <environments default="development">

12 <environment id="development">

13 <transactionManager type="JDBC" />

14 <!-- 配置数据库连接信息 -->

15 <dataSource type="POOLED">

16 <!-- value属性值引用db.properties配置文件中配置的值 -->

17 <property name="driver" value="${driver}" />

18 <property name="url" value="${url}" />

19 <property name="username" value="${name}" />

20 <property name="password" value="${password}" />

21 </dataSource>

22 </environment>

23 </environments>

24

25 </configuration>

[复制代码](javascript:void(0);)

**二、为实体类定义别名，简化sql映射xml文件中的引用**

　　之前，我们在sql映射xml文件中的引用实体类时，需要写上实体类的全类名(包名+类名)，如下：

<!-- 创建用户(Create) -->

<insert id="addUser" parameterType="**me.gacl.domain.User**">

insert into users(name,age) values(#{name},#{age})

</insert>

　　parameterType="**me.gacl.domain.User**"这里写的实体类User的全类名**me.gacl.domain.User，**每次都写这么一长串内容挺麻烦的，而我们希望能够简写成下面的形式

<insert id="addUser2" parameterType="**\_User**">

insert into users(name,age) values(#{name},#{age})

</insert>

　　parameterType="**\_User**"这样写就简单多了，为了达到这种效果，我们需要在conf.xml文件中为实体类="**me.gacl.domain.User**"定义一个别名为"**\_User**"，具体做法如下：  
　　在conf.xml文件中<configuration></configuration>标签中添加如下配置：

<typeAliases>

<typeAlias type="me.gacl.domain.User" alias="\_User"/>

</typeAliases>

　　这样就可以为**me.gacl.domain.User**类定义了一个别名为**\_User，**以后**\_User**就代表了**me.gacl.domain.User**类，这样sql映射xml文件中的凡是需要引用**me.gacl.domain.User**类的地方都可以使用**\_User**来代替，这就达到了一个简化实体类引用的目的。

　　除了可以使用<typeAlias type="me.gacl.domain.User" alias="\_User"/>这种方式单独为某一个实体类设置别名之外，我们还可以使用如下的方式批量为某个包下的所有实体类设置别名，如下：

[复制代码](javascript:void(0);)

<!-- 配置实体类的别名，配置实体类别名的目的是为了在引用实体类时可以使用实体类的别名来代替实体类，达到简写的目的 -->

<typeAliases>

<!-- 为实体类me.gacl.domain.User配置一个别名\_User -->

<!-- <typeAlias type="me.gacl.domain.User" alias="\_User"/> -->

<!-- 为me.gacl.domain包下的所有实体类配置别名，MyBatis默认的设置别名的方式就是去除类所在的包后的简单的类名

比如me.gacl.domain.User这个实体类的别名就会被设置成User

-->

<package name="me.gacl.domain"/>

</typeAliases>

[复制代码](javascript:void(0);)

　　<package name="me.gacl.domain"/>就表示为这个包下面的所有实体类设置别名。MyBatis默认的设置别名的方式就是去除类所在的包后的简单的类名，比如me.gacl.domain.User这个实体类的别名就会被设置成User。