

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration
PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
    <properties resource="jdbc.properties"/>

```

```

<!--

```

别名，在mapper映射中可以直接使用，例如：下面需要传入类型 Student，如果不用别名，那么久需要写下完整的包名+类名

```

<insert id="add" parameterType="Student">
    insert into t_student values(null,#{name},#{age})
</insert>

```

在mybatis中 java基本数据类型，int，byte等，以及自带的一些类型如 集合，ArrayList，map等有一个映射方式

在 org.apache.ibatis.type.TypeAliasRegistry 类中进行了注册，下面的图片附有映射

```

-->

```

```

<typeAliases>

```

```

<!-- 第一种方式，一个别名对应一个类名 -->

```

```

<typeAlias alias="Student" type="com.java1234.model.Student"/>

```

<!-- 第二种方式，这个包下面的所有类都会进行别名映射，那么上面的 Student 别名，就会自动变成 package 配置的包名+Student -->

```

<package name="com.java1234.model"/>

```

```

</typeAliases>

```

<!-- 环境，可以配置多个，然后选择一个默认的，一般情况下 分为 开发，测试，正式 三种 配置 -->

```

<environments default="development">

```

```

<!-- 开发 -->

```

```

<environment id="development">

```

<!-- 分为JDBC和Mannager两种：

JDBC： 应用程序负责管理数据库连接的生命周期。

Tomcat就选择JDBC

MANAGED： 由应用服务器负责管理数据库连接的生命周期；（一般商业服务器才有此功能，如JBOSS,WebLogic）-->

<transactionManager type="JDBC" />

<!-- 数据源的方式：类型有 UNPOOLED， POOLED， JNDI

UNPOOLED， 没有连接池，每次数据库操作， MyBatis都会创建一个新的连接，用完后，关闭；适合小并发项目；

POOLED， 用上了连接池；

JNDI， 使用应用服务器配置JNDI数据源获取数据库连接；

-->

<dataSource type="POOLED">

<property name="driver"

value="\${jdbc.driverClassName}" />

<property name="url" value="\${jdbc.url}" />

<property name="username" value="\${jdbc.username}" />

<property name="password" value="\${jdbc.password}" />

</dataSource>

</environment>

</environments>

<!-- 配置mybatis Mapper映射 -->

<mappers>

<!-- 一个一个的引入Mapper映射方式 -->

<mapper resource="com/java1234/mappers/StudentMapper.xml" />

<!-- 扫描的方式 -->

<package name="com.java1234.mappers"/>

</mappers>

</configuration>

别名	对应的 java 类型
_byte	byte
_long	long
_short	short
_int	int
_integer	int
_double	double
_float	float
_boolean	boolean
string	String
Byte	Byte
Long	Long
short	Short
Int	Integer
integer	Integer
double	Double
Float	Float
boolean	Boolean
Date	Date
decimal	BigDecimal

bigdecimal	BigDecimal
object	Object
Map	Map
hashmap	HashMap
List	List
arraylist	ArrayList
collection	Collection
iterator	Iterator