# 环境搭建

## 1）创建表

**CREATE** **TABLE** `**user**` **(**

`id` **int(**11**)** **NOT** **NULL** auto\_increment**,**

`username` **varchar(**32**)** **NOT** **NULL** **COMMENT** '用户名称'**,**

`birthday` **datetime** **default** **NULL** **COMMENT** '生日'**,**

`sex` **char(**1**)** **default** **NULL** **COMMENT** '性别'**,**

`address` **varchar(**256**)** **default** **NULL** **COMMENT** '地址'**,**

**PRIMARY** **KEY** **(**`id`**)**

**)** ENGINE**=**InnoDB **DEFAULT** CHARSET**=**utf8**;**

## 2）插入数据

**insert** **into** `**user**`**(**`id`**,**`username`**,**`birthday`**,**`sex`**,**`address`**)**

**values** **(**41**,**'老王'**,**'2018-02-27 17:47:08'**,**'男'**,**'北京'**),(**42**,**'小二王'**,**'2018-03-02 15:09:37'**,**'女'**,**'北京金燕龙'**),(**43**,**'小二王'**,**'2018-03-04 11:34:34'**,**'女'**,**'北京金燕龙'**),(**45**,**'传智播客'**,**'2018-03-04 12:04:06'**,**'男'**,**'北京金燕龙'**),(**46**,**'老王'**,**'2018-03-07 17:37:26'**,**'男'**,**'北京'**),(**48**,**'小马宝莉'**,**'2018-03-08 11:44:00'**,**'女'**,**'北京修正'**);**

## 3）创建Maven工程

<dependencies>  
 <dependency>  
 <groupId>org.mybatis</groupId>  
 <artifactId>mybatis</artifactId>  
 <version>3.4.5</version>  
 </dependency>  
 <dependency>  
 <groupId>mysql</groupId>  
 <artifactId>mysql-connector-java</artifactId>  
 <version>5.1.6</version>  
 </dependency>  
 <dependency>  
 <groupId>log4j</groupId>  
 <artifactId>log4j</artifactId>  
 <version>1.2.12</version>  
 </dependency>  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.10</version>  
 </dependency>

## 4）创建实体类

public class User implements Serializable {  
 private Integer id**;** private String username**;** private Date birthday**;** private String address**;** public Integer getId() {  
 return id**;** }  
  
 public void setId(Integer id) {  
 this.id = id**;** }  
  
 public String getUsername() {  
 return username**;** }  
  
 public void setUsername(String username) {  
 this.username = username**;** }  
  
 public Date getBirthday() {  
 return birthday**;** }  
  
 public void setBirthday(Date birthday) {  
 this.birthday = birthday**;** }  
  
 public String getAddress() {  
 return address**;** }  
  
 public void setAddress(String address) {  
 this.address = address**;** }  
  
 @Override  
 public String toString() {  
 return "User{" +  
 "id=" + id +  
 ", username='" + username + '\'' +  
 ", birthday=" + birthday +  
 ", address='" + address + '\'' +  
 '}'**;** }  
}

## 5）创建dao接口IUserDao

/\*  
用户的持久层接口  
 \*/  
public interface IUserDao {  
 */\*\*  
 \* 查询所有的操作  
 \** ***@return*** *\*/* List<User> findAll()**;**}

## 6）创建Mybatis的主配置文件（先导入约束—在config.xml中）

<?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">  
<!--Mybatis的主配置文件-->  
<configuration>  
 <!--配置环境-->  
 <environments default="mysql">  
 <!--配置mysql的环境-->  
 <environment id="mysql">  
 <!--配置事务类型-->  
 <transactionManager type="jdbc"></transactionManager>  
 <!--配置数据源（连接池）-->  
 <dataSource type="POOLED">  
 <!--配置连接数据库的4个基本信息-->  
 <property name="driver" value="com.mysql.jdbc.Driver"/>  
 <property name="url" value="jdbc:mysql://localhost:3306/eesy\_mybatis"/>  
 <property name="username" value="root"/>  
 <property name="password" value="root"/>  
 </dataSource>  
 </environment>  
 </environments>  
  
 <!--指定映射配置文件的位置,映射配置文件指的是每个dao独立的配置文件-->  
 <mappers>  
 <mapper resource="com/itheima/dao/IUserDao.xml"/>  
 </mappers>  
</configuration>

## 7）创建映射文件IUserDao.xml（也要先导入约束 mapper.xml）

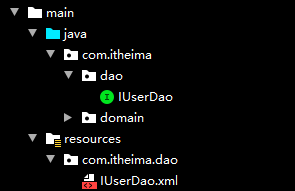
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE mapper  
 PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
 <mapper namespace="com.itheima.dao.IUserDao">  
 <!--配置查询所有-->  
 <select id="findAll">  
 select \* from user  
 </select>  
</mapper>

## 8）并且直接把Log4J 的配置文件Copy到Resources目录下

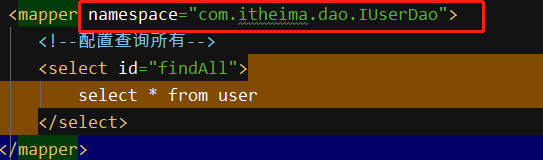
## 9）在test目录下创建包和类:com.itheima.test.Mybatis

## 10）注意事项

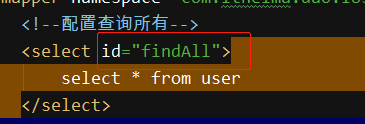
1.映射配置文件的位置必须和dao接口文件的包结构相同



1. 映射配置文件的namespace必须是IUserDao接口的全限定类型（包名.类名）



1. 映射配置文件的id





当我们实现了以上几点，就无需再书写dao的是实现类了，接下来的将有Mybatis来帮我们实现。

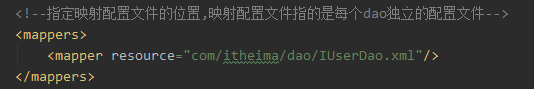
# 2.自定义Mybatis的分析

## 1）执行查询所有的分析

1.有了连接数据库的信息，就能创建Connection对象

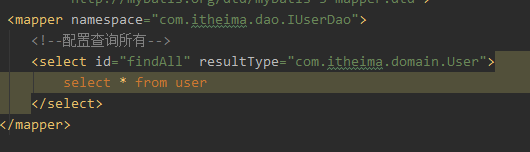


2.映射配置信息



3.有了sql语句，就能把sql传入PrepareStatement,执行sql语句

此配置中还有封装的实体类的全限定类名



1. 读取dom4j解析xml的技术来读取配置文件
2. 我们在得到封装的实体类对象的全限定类名后，可以根据反射，来获取对象E



6.使用动态代理创建代理对象来实现对于原来对象的增强，调用selectList方法



以下是全景图：

