conda activate python37

清华：<https://pypi.tuna.tsinghua.edu.cn/simple>

阿里云：<http://mirrors.aliyun.com/pypi/simple/>

中国科技大学 <https://pypi.mirrors.ustc.edu.cn/simple/>

华中理工大学：<http://pypi.hustunique.com/>

山东理工大学：<http://pypi.sdutlinux.org/>

豆瓣：<http://pypi.douban.com/simple/>

Springxml配置

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

<beans xmlns=*"http://www.springframework.org/schema/beans"*

xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xmlns:context=*"http://www.springframework.org/schema/context"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans.xsd*

*"*>

<bean id=*"dataSource"* class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"com.mysql.jdbc.Driver"*/>

<property name=*"url"* value=*"jdbc:mysql://localhost:3306/千万级数据插入?useUnicode=true&amp;useSSL=true&amp;characterEncoding=gbk"*/>

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

<property name=*"password"* value=*"root"*/>

</bean>

<bean id=*"jdbcTemplate"* class=*"org.springframework.jdbc.core.JdbcTemplate"*>

<property name=*"dataSource"* ref=*"dataSource"*/></bean>

<bean id=*"accountDao"* class=*"lab3.accountdao"*>

<property name=*"jdbcTemplate"* ref=*"jdbcTemplate"*/>

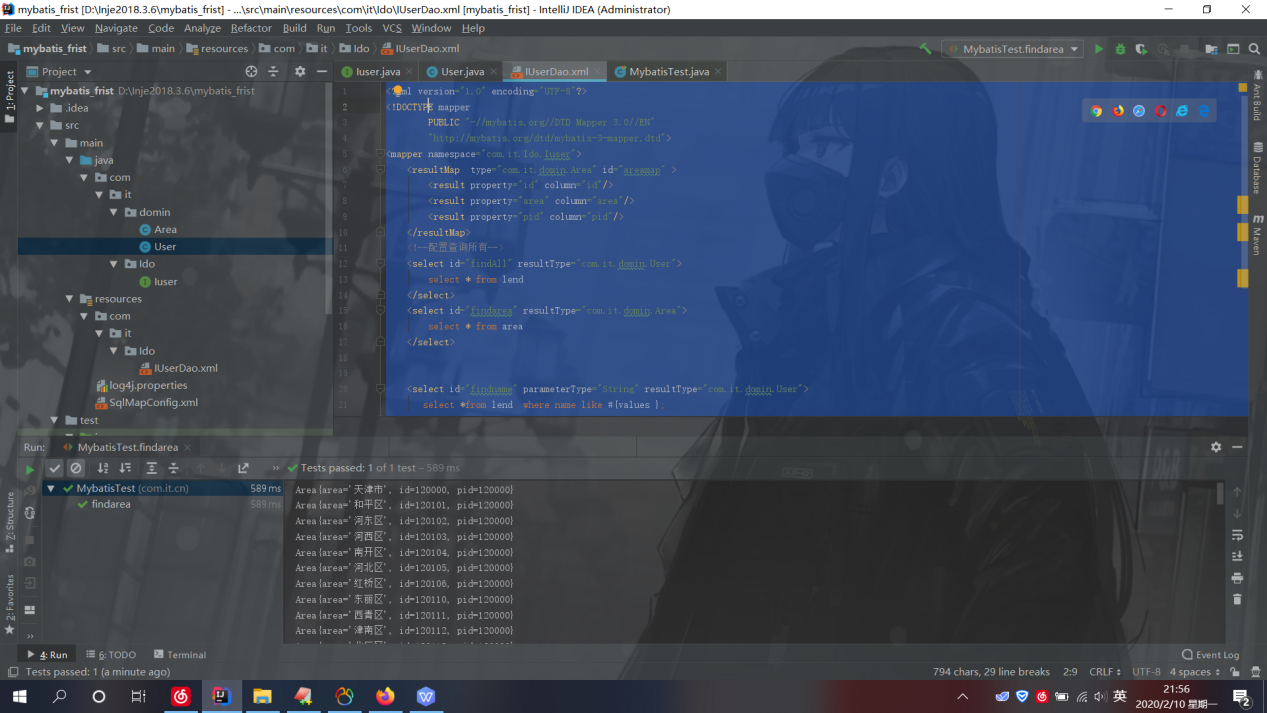
</bean>

</beans>

Mybatis核心配置

<?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.it.Ido.Iuser">  
 <resultMap type="com.it.domin.Area" id="areamap" >  
 <result property="id" column="id"/>  
 <result property="area" column="area"/>  
 <result property="pid" column="pid"/>  
 </resultMap>  
 <!--配置查询所有-->  
 <select id="findAll" resultType="com.it.domin.User">  
 select \* from lend  
 </select>  
 <select id="findarea" resultType="com.it.domin.Area">  
 select \* from area  
 </select>  
  
  
 <select id="findname" parameterType="String" resultType="com.it.domin.User">  
 select \*from lend where name like #{values };  
 </select>  
  
  
  
  
  
  
  
</mapper>

Mybatis的动态sql配置



Mybatis的动态sql语句

<?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.it.Ido.Iuser">  
 <resultMap type="com.it.domin.Area" id="areamap" >  
 <id property="id" column="id"/>  
 <result property="area" column="area"/>  
 <result property="pid" column="pid"/>  
  
 </resultMap>  
 <!--配置查询所有-->  
  
 <select id="findarea" resultMap="areamap">  
 select \* from area  
 </select>  
 <select id="findAll" resultType="com.it.domin.User">  
 select \*from user  
 </select>  
 <select id="findbycondtion" parameterType="com.it.domin.User" resultType="com.it.domin.User">  
 select \*from user  
 <where>  
 <if test="username!=null and address!=null">  
 and username like #{username}  
 </if>  
  
 </where>  
  
 </select>  
 <select id="findwhere" parameterType="com.it.domin.Quervo" resultType="com.it.domin.User">  
 SELECT \*FROM user  
 <where>  
 <if test="name!=null and name.size()>0">  
 <foreach collection="name" open="and username in(" close=")" item="username" separator=",">  
 #{username}  
 </foreach>  
 </if>  
 </where>  
  
  
  
 </select>  
  
  
  
  
  
  
  
  
  
</mapper>

测试类

package com.it.cn;  
  
//import com.it.domin.User;  
import com.it.Ido.Iuser;  
import com.it.domin.Area;  
import com.it.domin.Quervo;  
import com.it.domin.User;  
import org.apache.ibatis.io.Resources;  
import org.apache.ibatis.session.SqlSession;  
import org.apache.ibatis.session.SqlSessionFactory;  
import org.apache.ibatis.session.SqlSessionFactoryBuilder;  
import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
  
import java.io.InputStream;  
import java.util.ArrayList;  
import java.util.List;  
  
*/\*\*  
 \** ***@author*** *黑马程序员  
 \** ***@Company*** *http://www.ithiema.com  
 \* mybatis的入门案例  
 \*/*public class MybatisTest {  
 InputStream in;  
 SqlSession session;  
 Iuser userDao;  
 @Before  
 public void init()throws Exception{  
 in = Resources.*getResourceAsStream*("SqlMapConfig.xml");  
 //2.创建SqlSessionFactory工厂  
 SqlSessionFactoryBuilder builder = new SqlSessionFactoryBuilder();  
 SqlSessionFactory factory = builder.build(in);  
 //3.使用工厂生产SqlSession对象  
 session = factory.openSession(true);  
 //4.使用SqlSession创建Dao接口的代理对象  
 userDao = session.getMapper(Iuser.class);  
 }  
 */\*\*  
 \* 入门案例  
 \** ***@param*** *args  
 \*/* @After  
 public void destroy()throws Exception{  
 session.close();  
 in.close();  
 }  
 @Test  
 public void testall()throws Exception {  
 //1.读取配置文件  
  
 //5.使用代理对象执行方法  
 List<User> users = userDao.findAll();  
 for(User user : users){  
 System.*out*.println(user);  
 }  
  
 //6.释放资源  
  
 }  
 @Test  
 public void findbyname(){  
 List<User> users=userDao.findname("%陈%");//模糊查询  
 // List<User> users=userDao.findname("陈森");  
 for(User user : users){  
 System.*out*.println(user);  
 }  
 }  
  
 @Test  
 public void findarea(){  
 List<Area> areas=userDao.findarea();  
 for(Area area:areas){  
 System.*out*.println(area);  
 }  
  
 }  
 @Test  
 public void findcondtion(){  
 User us=new User();  
 us.setUsername("%谢%");  
 us.setAddress("%连云港%");  
  
 List<User> users=userDao.findbycondtion(us);  
 for(User user:users ){  
 System.*out*.println(user);}  
  
 }  
 @Test  
 public void show(){  
 List<String> list=new ArrayList<String>();  
 list.add("谢凌薇");  
 list.add("宁紫陌");  
  
 Quervo vo=new Quervo();  
 vo.setName(list);  
  
 List<User> users=userDao.findwhere(vo);  
 for(User user:users ){  
 System.*out*.println(user);}  
  
 }  
  
  
  
}