

案例3-1 基于Mybatis实现用户登录与注册功能

一、实验目标

- 1、熟悉IDEA开发环境的使用及数据库管理工具的使用
- 2、掌握Mybatis和Druid框架的使用
- 3、掌握Mybatis Gernator的使用
- 4、使用注解的方式整合MyBatis
- 5、使用配置文件的方式整合MyBatis

二、实验环境

Windows操作系统+JDK1.8+Maven3.6+IDEA2019+Mysql数据库+Mybatis Gernator

三、实验要求

- 1、准备好Mysql数据库环境:mysql5.5+navicat
- 2、使用IDEA的Spring Initializr构建Spring Boot项目并添加Spring web、Mybatis和Druid框架
- 3、在全局配置文件中配置Mybatis参数
- 4、Mybatis Gernator生成Mybatis数据库访问接口层
- 5、在浏览器上访问 <http://localhost:8080/api/login?name=test&password=1234>时在客户端提示并在后服务端控制台输出：用户名称或密码是否正确
- 6、<http://localhost:8080/api/add?id=1&name=test&password=1234>时在客户端提示并在服务端控制台输出：用户信息新增是否成功
- 7、<http://localhost:8080/api/update?id=1&name=test&password=test>时在客户端提示并在服务端控制台输出：用户信息更新是否成功
- 8、<http://localhost:8080/api/delete?id=1>时在客户端提示并在服务端控制台输出：用户信息删除是否成功
- 1、

四、实验关键步骤

- 1、启动数据库服务并创建好数据库 (grade)及表：用户信息表t_user(用户id,用户名称, 用户密码,手机, 邮箱,头像)。

名	类型	长度	小数点	允许空值	
id	int			<input type="checkbox"/>	1
username	varchar	50		<input type="checkbox"/>	
password	varchar	50		<input type="checkbox"/>	
telephone	varchar	50		<input checked="" type="checkbox"/>	
email	varchar	200		<input checked="" type="checkbox"/>	
photo	varchar	200		<input checked="" type="checkbox"/>	

默认:

注释:

字符集: **utf8**

排序规则:

键长度:

☐ 二进制

```
CREATE database if NOT EXISTS `grade` default character set utf8mb4 collate
utf8mb4_unicode_ci;
use `grade`;
DROP TABLE IF EXISTS `t_user`;
CREATE TABLE `t_user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  `telephone` varchar(50) DEFAULT NULL,
  `email` varchar(200) DEFAULT NULL,
  `photo` varchar(200) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

2、使用Spring Initializr方式构建Spring Boot项目并选择 需要用到的框架，并添加MySQL数据库连接驱动。

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.40</version>
  <scope>runtime</scope>
</dependency>
<dependency>
  <groupId>com.alibaba</groupId>
  <artifactId>druid-spring-boot-starter</artifactId>
  <version>1.1.10</version>
</dependency>
```

```
spring.datasource.url=jdbc:mysql://localhost:3306/yc??characterEncoding=UTF-8&useUnicode=true&useSSL=false
spring.datasource.username=root
spring.datasource.password=123456
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
```

```
# 5.x的mysql数据库驱动:
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/grade??characterEncoding=UTF-8&useUnicode=true&useSSL=false
spring.datasource.username=root
spring.datasource.password=
```

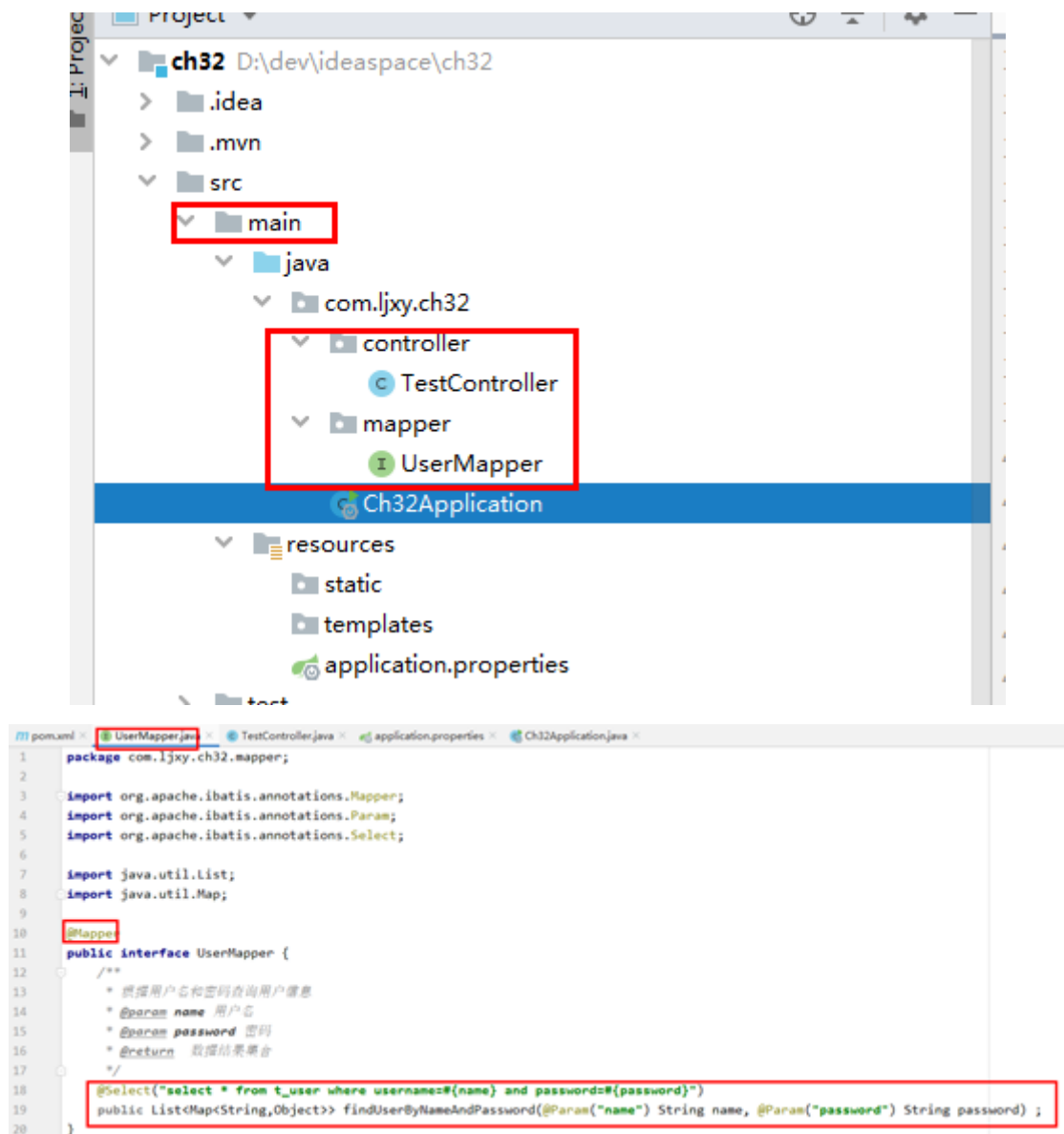
如果是用的MySQL数据库连接驱动8.0以上的:

```
<dependency>
  <groupId>org.mybatis.spring.boot</groupId>
  <artifactId>mybatis-spring-boot-starter</artifactId>
  <version>2.1.4</version>
</dependency>
<!-- 没指定版本即为springboot中的默认mysql驱动是8以上的驱动-->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <scope>runtime</scope>
</dependency>
```

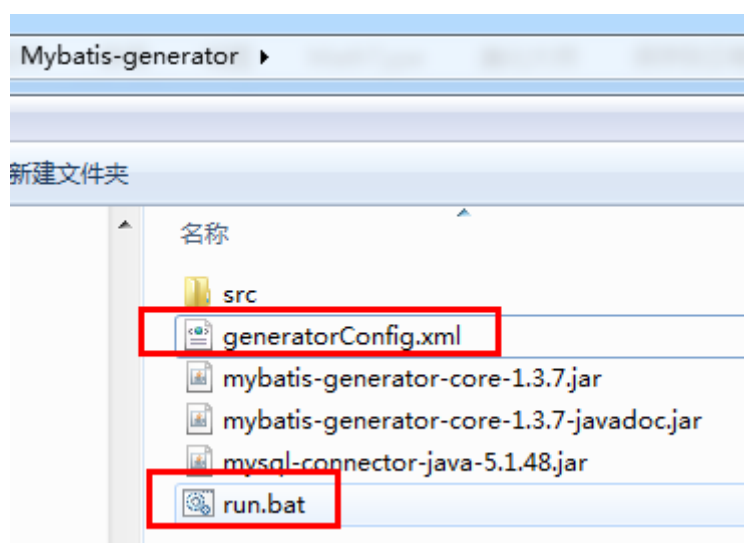
此时数据库驱动的配置为:

```
# 8.x的mysql数据库驱动:
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
# 数据源名称
spring.datasource.name=defaultDataSource
# 数据库连接地址 UTC代表的是全球标准时间 , 但是我们使用的时间是北京时区也就是东八区, 领先UTC八个小时。 使用上海时间 serverTimezone=Asia/Shanghai 北京时间 GMT%2B8
spring.datasource.url=jdbc:mysql://localhost:3306/grade?
serverTimezone=Asia/Shanghai&useSSL=false&characterEncoding=utf8
# 数据库用户名&密码:
spring.datasource.username=root
spring.datasource.password=
```

3、创建Mapper接口和控制器类



4、Mybatis Gernator生成Mybatis数据库访问接口层（注意一定要有src目录，否则不会生成src），利用Mybatis-generator工具自动生成Mybatis模块，并复制到项目中。



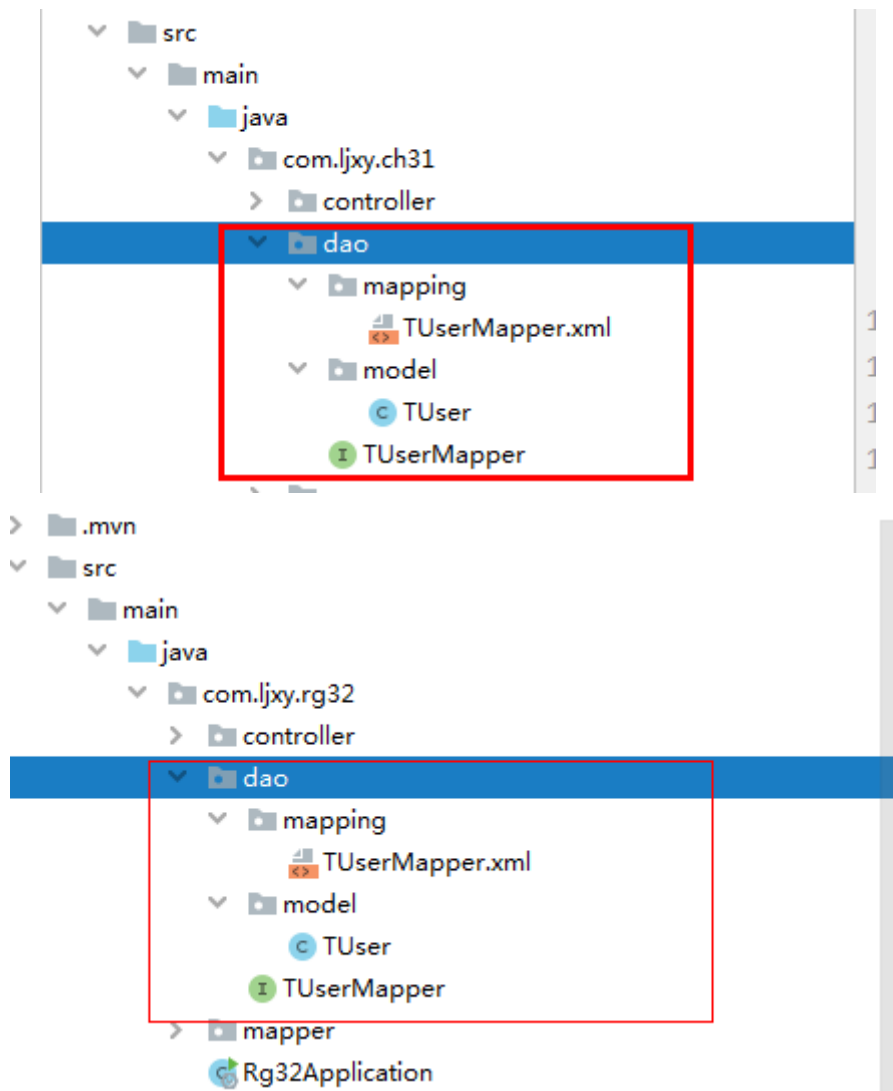
generatorConfig.xml中参数配置：

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE generatorConfiguration PUBLIC "-//mybatis.org/DTD Mybatis Generator Configuration 1.0/EN" "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">
3 <generatorConfiguration>
4   <!-- 数据库驱动包位置 -->
5   <classPathEntry location=".\mybatis-generator-java-1.3.5.jar" />
6
7   <context id="DB2Tables" targetRuntime="MyBatis3">
8     <commentGenerator>
9       <property name="suppressAllComments" value="true" />
10    </commentGenerator>
11
12    <!-- 数据库连接URL、用户名、密码 -->
13    <jdbcConnection driverClass="com.mysql.jdbc.Driver" connectionURL="jdbc:mysql://localhost:3306/test" userId="root" password="" -->
14    <jdbcConnection driverClass="com.mysql.jdbc.Driver" connectionURL="jdbc:mysql://127.0.0.1:3306/test" userId="root" password="123456" />
15  </jdbcConnection>
16
17  <javaTypeResolver>
18    <property name="forcedBigDecimals" value="false" />
19  </javaTypeResolver>
20
21  <!-- 生成模型的包名和位置 -->
22  <javaModelGenerator targetPackage="com.ljxy.ch31.dao.model" targetProject=".\src">
23    <property name="enableSubPackages" value="true" />
24    <property name="trimStrings" value="true" />
25  </javaModelGenerator>
26
27  <!-- 生成映射文件的包名和位置 -->
28  <sqlMapGenerator targetPackage="com.ljxy.ch31.dao.mapping" targetProject=".\src">
29    <property name="enableSubPackages" value="true" />
30  </sqlMapGenerator>
31
32  <!-- 生成DAO的包名和位置 -->
33  <javaClientGenerator type="JDBCMAPPER" targetPackage="com.ljxy.ch31.dao" targetProject=".\src">
34    <property name="enableSubPackages" value="true" />
35  </javaClientGenerator>
36
37  <!-- 要生成和包含的表(要包含的表可以不写，表名对类名) -->
38  <table tableName="t_user" domainObjectName="TUser" enableCountByExample="false" enableUpdateByExample="false" enableDeleteByExample="false" enableSelectByExample="false" />
39 </context>
40 </generatorConfiguration>

```

先修改generatorConfig.xml的数据库配置参数，然后点击run.bat运行即生成Mybatis模块文件：



1. 使用注解@Mapper配置TUserMapper 或 在启动类前配置：@MapperScan("com.ljxy.ch31.dao")

```

1 package com.ljxy.ch31.dao;
2
3 import com.ljxy.ch31.dao.model.TUser;
4 import org.apache.ibatis.annotations.Mapper;
5 import org.apache.ibatis.annotations.Param;
6 import org.apache.ibatis.annotations.Select;
7
8 import java.util.List;
9 import java.util.Map;
10
11 @Mapper
12 public interface TUserMapper {
13     int deleteByPrimaryKey(Integer id);
14 }

```

2. 在全局配置文件设置mybatis别名:

```

#指定Mybatis的Mapper文件
mybatis.mapperLocations=classpath:*/mapping/*.xml
#指定Mybatis的实体目录
mybatis.typeAliasesPackage=com.ljxy.ch31.dao.model

```

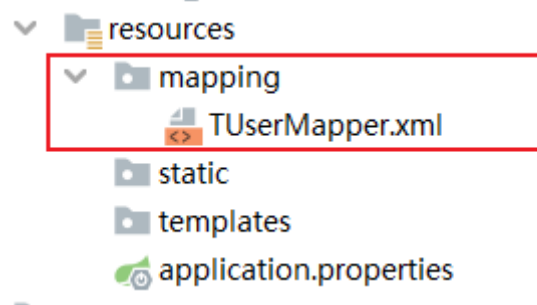
或

```

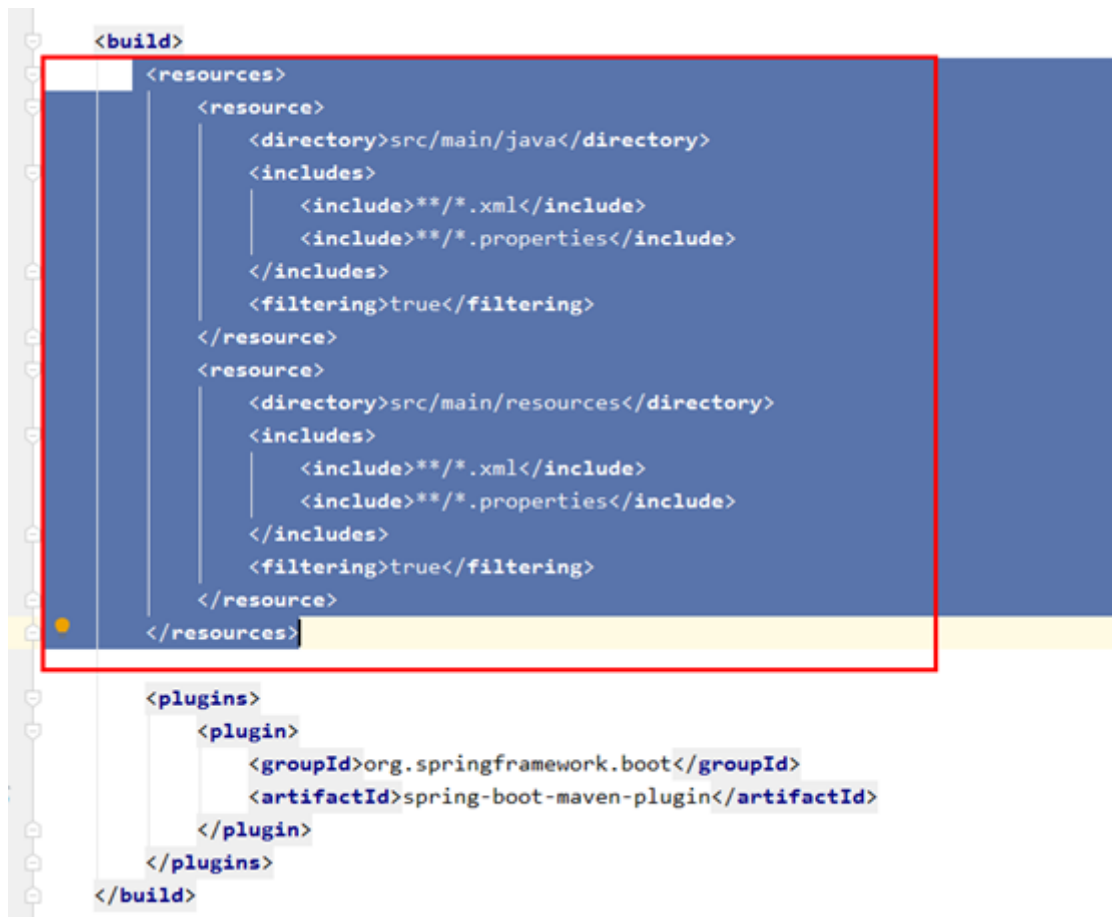
#指定Mybatis的Mapper文件
mybatis.mapper-locations=classpath:*/mapping/*.xml
#指定Mybatis的实体目录
mybatis.type-aliases-package=com.ljxy.ch31.dao.model

```

3. 在pom文件配置resource:



若将mapping目录移到resources下即可省去下面的配置;否则需要按下面的方法进行配置:



```
<resources>
  <resource>
    <directory>src/main/java</directory>
    <includes>
      <include>**/*.xml</include>
      <include>**/*.properties</include>
    </includes>
    <filtering>true</filtering>
  </resource>
  <resource>
    <directory>src/main/resources</directory>
    <includes>
      <include>**/*.xml</include>
      <include>**/*.properties</include>
    </includes>
    <filtering>true</filtering>
  </resource>
</resources>
```

4. 基于模块实现数据库访问和创建测试类

```

import java.util.HashMap;
import java.util.List;
import java.util.Map;

@RestController
public class TestController {
    @Autowired
    UserMapper userMapper;

    @Autowired
    TUserMapper tUserMapper;

    @RequestMapping("/update")
    public String update(TUser user){
        System.out.println(user.toString());
        try {
            int i= tUserMapper.updateByPrimaryKeySelective(user);
            if (i>0){
                return "更新成功。";
            }else if (i==0){
                return "无此数据,不需更新";
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return "更新失败。";
    }

    @RequestMapping("/find")
    public String find(TUser user) {
        TUser user1 = tUserMapper.selectByPrimaryKey(user.getId());
        return user1.toString();
    }
}

```

若出现下面的提示：

java.lang.IllegalStateException: Optional int parameter 'id' is present but cannot be translated into a null value due to being declared as a primitive type. Consider declaring it as object wrapper for the corresponding primitive type.

则需要将int id改为：Integer id


```

50 //6、 http://localhost:8080/ api/add?id=1&name=test&password=1234
51 @RequestMapping("/add")
52 public String add(int id,String name ,String password){
53     try {
54         tUserMapper.insert(new TUser(id,name,password));
55     } catch (Exception e) {
56         e.printStackTrace();
57         return "新增失败。";
58     }
59     return "新增成功。";
60 }
61
62 @RequestMapping("/delete")
63 public String delete(int id){
64     try {
65         int i= tUserMapper.deleteByPrimaryKey(id);
66         if (i>0){
67             return "删除成功。";
68         }else if (i==0){
69             return "无此数据。";
70         }
71     } catch (Exception e) {
72         e.printStackTrace();
73     }
74     return "删除失败。";
75 }

```

5. 启动应用后在浏览器中查看运行结果。

一、 实验结果及测试效果截图