

版权声明：本文为博主原创文章，未经博主允许不得转载。

## 基本准备工作

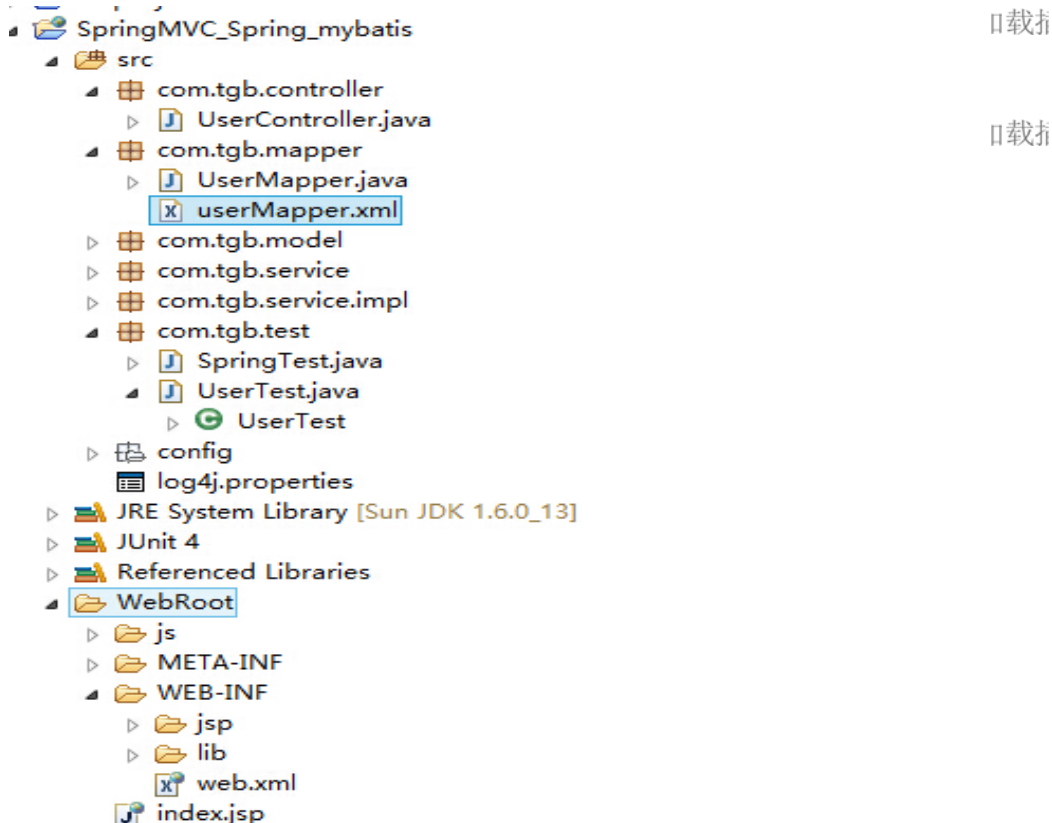
- 当然，这些jar还不够，还需要MySQL数据库与驱动，log4j的jar等等。下面我们开始今天的旅行：

## 第一步：创建数据库表

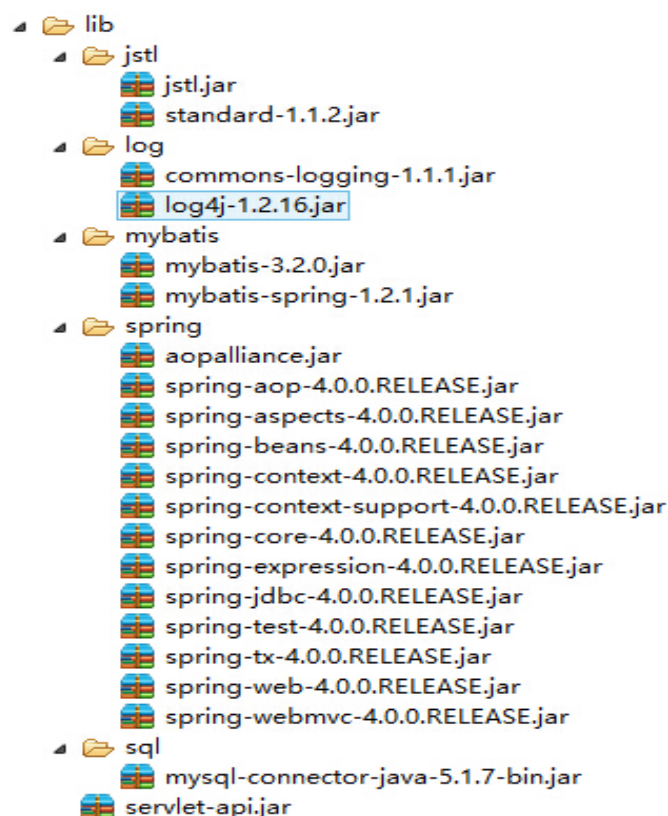
在Navicat下执行如下sql命令创建数据库mybatis和表t\_user

我们先看一下项目的完整目录，再继续下面的内容

日载在



## 第二步：添加jar包



对于下面代码的内容, 我们就不再一一贴出来, 只是把最重要的内容贴出来, 大家可以下载源码。

## 第三步：创建model

创建一个model包并在其下创建一个User.Java文件。

```
[java] view plain copy print ? C 8
01. package com.tgb.model;
02.
03. /**
04.  * 用户
05.  * @author liang
06.  *
07.  */
08. public class User {
09.
10.     private int id;
11.     private String age;
12.     private String userName;
13.     public User(){
14.         super();
15.     }
16.     public int getId() {
17.         return id;
18.     }
19.     public void setId(int id) {
20.         this.id = id;
21.     }
22.     public String getAge() {
23.         return age;
24.     }
25.     public void setAge(String age) {
26.         this.age = age;
27.     }
28.     public String getUserName() {
29.         return userName;
30.     }
31.     public void setUserName(String userName) {
32.         this.userName = userName;
33.     }
34.     public User(int id, String age, String userName) {
35.         super();
36.         this.id = id;
37.         this.age = age;
38.         this.userName = userName;
39.     }
40. }
```

第四步：创建DAO接口

创建一个包mapper，并在其下创建一个UserMapper.java文件作为DAO接口。

```
[java] view plain copy print ? C 8
01. package com.tgb.mapper;
```

```

02.
03. import java.util.List;
04.
05. import com.tgb.model.User;
06.
07. public interface UserMapper {
08.
09.     void save(User user);
10.     boolean update(User user);
11.     boolean delete(int id);
12.     User findById(int id);
13.     List<User> findAll();
14. }

```

[下载](#)

## 第五步：实现DAO接口

在dao包下创建一个UserMapper.xml文件作为上一步创建的DAO接口的实现。

```

[html] view plain copy print ?
01. <?xml version="1.0" encoding="UTF-8"?>
02. <!DOCTYPE mapper PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN"
03. "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
04. <!--
05.     namespace: 必须与对应的接口全类名一致
06.     id: 必须与对应接口的某个对应的方法名一致
07.
08. -->
09. <mapper namespace="com.tgb.mapper.UserMapper">
10.
11.     <insert id="save" parameterType="User">
12.         insert into t_user(user_name,user_age) values(#{userName},#{age})
13.     </insert>
14.
15.     <update id="update" parameterType="User">
16.         update t_user set user_name=#{userName},user_age=#{age} where user_id=#{id}
17.     </update>
18.
19.     <delete id="delete" parameterType="int">
20.         delete from t_user where user_id=#{id}
21.     </delete>
22.
23.     <!-- mybatis_config中配置的alias类别名,也可直接配置resultType为类路径 -->
24.     <select id="findById" parameterType="int" resultType="User">
25.         select user_id id,user_name userName,user_age age from t_user where user_id=#{
{id}
26.     </select>
27.
28.     <select id="findAll" resultType="User">
29.         select user_id id,user_name userName,user_age age from t_user
30.     </select>
31.

```

[下载](#)

32. &lt;/mapper&gt;

这里对这个xml文件作几点说明：

- 1、namespace必须与对应的接口全类名一致。
- 2、id必须与对应接口的某个对应的方法名一致即必须要和UserMapper.java接口中的方法同名。

## 第六步：Mybatis和Spring的整合

对于Mybatis和Spring的整合是这篇博文的重点，需要配置的内容在下面有详细的解释。

[html] view plain copy print ?

```

01. <?xml version="1.0" encoding="UTF-8"?>
02. <beans xmlns="http://www.springframework.org/schema/beans"
03.     xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xmlns:p="http://www.springframework.org/schema/p"
04.     xmlns:context="http://www.springframework.org/schema/context" xmlns:tx="http://www.spr
05.     xsi:schemaLocation="
06.         http://www.springframework.org/schema/beans
07.         http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
08.         http://www.springframework.org/schema/context
09.         http://www.springframework.org/schema/context/spring-context-4.0.xsd
10.         http://www.springframework.org/schema/tx
11.         http://www.springframework.org/schema/tx/spring-tx-4.0.xsd">
12.
13.     <!-- 1. 数据源 : DriverManagerDataSource -->
14.     <bean id="dataSource"
15.         class="org.springframework.jdbc.datasource.DriverManagerDataSource">
16.         <property name="driverClassName" value="com.mysql.jdbc.Driver" />
17.         <property name="url" value="jdbc:mysql://localhost:3306/mybatis" />
18.         <property name="username" value="root" />
19.         <property name="password" value="123456" />
20.     </bean>
21.
22.     <!--
23.         2. mybatis的SqlSession的工厂: SqlSessionFactoryBean dataSource:引用数据源
24.
25.         MyBatis定义数据源,同意加载配置
26.     -->
27.     <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
28.         <property name="dataSource" ref="dataSource"></property>
29.         <property name="configLocation" value="classpath:config/mybatis-config.xml" />
30.     </bean>
31.
32.     <!--
33.         3. mybatis自动扫描加载Sql映射文件/接
34.         口 : MapperScannerConfigurer sqlSessionFactory
35.
36.         basePackage:指定sql映射文件/接口所在的包（自动扫描）

```

```
36.      -->
37.      <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
38.          <property name="basePackage" value="com.tgb.mapper"></property>
39.          <property name="sqlSessionFactory" ref="sqlSessionFactory"></property>
40.      </bean>
41.
42.      <!--
43.          4. 事务管理 : DataSourceTransactionManager dataSource:引用上面定义的数据源
44.      -->
45.      <bean id="txManager"
46.          class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
47.          <property name="dataSource" ref="dataSource"></property>
48.      </bean>
49.
50.      <!-- 5. 使用声明式事务
51.          transaction-manager: 引用上面定义的事务管理器
52.      -->
53.      <tx:annotation-driven transaction-manager="txManager" />
54.
55. </beans>
```

## 第七步：mybatis的配置文件

```
[html] view plain copy print ? C 8
01. <?xml version="1.0" encoding="UTF-8"?>
02. <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
03. "http://mybatis.org/dtd/mybatis-3-config.dtd">
04. <configuration>
05.
06.     <!-- 实体类,简称 -设置别名 -->
07.     <typeAliases>
08.         <typeAlias alias="User" type="com.tgb.model.User" />
09.     </typeAliases>
10.     <!-- 实体接口映射资源 -->
11.     <!--
12.         说明：如果xxMapper.xml配置文件放在和xxMapper.java统一目录下，mappers也可以省略，因为
13.         org.mybatis.spring.mapper.MapperFactoryBean默认会去查找与xxMapper.java相同目录和名称的
14.         xxMapper.xml
15.     -->
16.     <mappers>
17.         <mapper resource="com/tgb/mapper/userMapper.xml" />
18.     </mappers>
19. </configuration>
```