■ 分类: 【Mybatis】(2) ▼

■版权声明:本文为博主原创文章,未经博主允许不得转载。

上篇博文我们介绍了mybatis的基本概念与原理,这篇博文我们通过**spring**与Mybatis集成,开发一个简单用户增删改查的Web项目。

基本准备工作

- 1、安装JDK1.6以上版本,安装与配置
- 2、下载mybatis-3.2.0版: https://repo1.maven.org/maven2/org/mybatis/mybatis/
- 3、下载mybatis-spring-1.2.1版: https://repo1.maven.org/maven2/org/mybatis/mybatis-spring/
- 4、Spring-4.0.0的版本
- 5、tomacat6.x以上版本即可

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

```
[sql] view plain copy print ? \subset \mathcal{V}
       CREATE DATABASE IF NOT EXISTS mybatis;
01.
       [sql] view plain copy print ? \subset \mathcal{V}
01.
       USE mybatis;
       [sql] view plain copy print ? \subset \mathcal{V}
       create table t_user
01.
02.
03.
           user_id int(11) NOT NULL AUTO_INCREMENT,
04.
           user_name varchar(20) not null,
05.
           user age varchar(20) not null,
           PRIMARY KEY (user_id)
06.
07.
      )ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

我们先看一下项目的完整目录,再继续下面的内容

口载抗

第二步:添加jar包

D 🗁 lib

x web.xml index.jsp

🛮 🗁 lib 🛮 🗁 jstl 💼 jstl.jar 📻 standard-1.1.2.jar 🛮 🗁 log 💼 commons-logging-1.1.1.jar 💼 log4j-1.2.16.jar mybatis 💼 mybatis-3.2.0.jar 💼 mybatis-spring-1.2.1.jar spring 💼 aopalliance.jar 💼 spring-aop-4.0.0.RELEASE.jar 💼 spring-aspects-4.0.0.RELEASE.jar 💼 spring-beans-4.0.0.RELEASE.jar 💼 spring-context-4.0.0.RELEASE.jar 💼 spring-context-support-4.0.0.RELEASE.jar 💼 spring-core-4.0.0.RELEASE.jar 💼 spring-expression-4.0.0.RELEASE.jar 💼 spring-jdbc-4.0.0.RELEASE.jar 💼 spring-test-4.0.0.RELEASE.jar 💼 spring-tx-4.0.0.RELEASE.jar 💼 spring-web-4.0.0.RELEASE.jar 💼 spring-webmvc-4.0.0.RELEASE.jar 🛮 🗁 sql 🚘 mysql-connector-java-5.1.7-bin.jar 🚌 servlet-api.jar

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。

```
[iava] view plain copy print ? 		 ♀
01.
      package com.tgb.model;
92.
      /**
03.
       * 用户
04.
       * @author liang
05.
06.
       */
07.
      public class User {
08.
09.
10.
          private int id;
11.
          private String age;
12.
          private String userName;
13.
          public User(){
14.
               super();
                                                                         口载抗
15.
          public int getId() {
16.
17.
               return id;
18.
          public void setId(int id) {
19.
20.
               this.id = id;
21.
22.
          public String getAge() {
23.
               return age;
24.
25.
          public void setAge(String age) {
26.
               this.age = age;
27.
          public String getUserName() {
28.
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();
               this.id = id;
36.
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 & package com.tgb.mapper;
```

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 ? C &
01.
      <?xml version="1.0" encoding="UTF-8"?>
02.
      <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"</pre>
      "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
03.
94.
          namespace: 必须与对应的接口全类名一致
05.
          id:必须与对应接口的某个对应的方法名一致
96.
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>
                                                                      日载右
14.
15.
          <update id="update" parameterType="User">
              update t user set user name=#{userName},user age=#{age} where user id=#{id}
16.
          </update>
17.
18.
          <delete id="delete" parameterType="int">
19.
20.
              delete from t_user where user_id=#{id}
21.
          </delete>
22.
23.
           <!-- mybsits_config中配置的alias类别名,也可直接配置resultType为类路劲 -->
          <select id="findById" parameterType="int" resultType="User">
24.
25.
              select user_id id,user_name userName,user_age age from t_user where user_id=#
      {id}
          </select>
26.
27.
          <select id="findAll" resultType="User">
28.
29.
              select user_id id,user_name userName,user_age age from t_user
30.
          </select>
```

```
32. </mapper>
```

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 ? C &
              <?xml version="1.0" encoding="UTF-8"?>
01.
02.
              <beans xmlns="http://www.springframework.org/schema/beans"</pre>
93
                       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.springframework.org/schema/context" xmlns:tx="http://www.springframework.org/
05.
                       xsi:schemaLocation="
                               http://www.springframework.org/schema/beans
96.
97.
                               http://www.springframework.org/schema/beans/spring-beans-4.0.xsd
                               http://www.springframework.org/schema/context
08.
                               http://www.springframework.org/schema/context/spring-context-4.0.xsd
99
                               http://www.springframework.org/schema/tx
10.
                               http://www.springframework.org/schema/tx/spring-tx-4.0.xsd">
11.
12.
                       <!-- 1. 数据源 : DriverManagerDataSource -->
13.
14.
                       <bean id="dataSource"</pre>
15.
                                class="org.springframework.jdbc.datasource.DriverManagerDataSource">
                                cproperty name="driverClassName" value="com.mysql.jdbc.Driver" />
16.
17.
                                roperty name="url" value="jdbc:mysql://localhost:3306/mybatis" />
                                roperty name="username" value="root" />
18.
                                roperty name="password" value="123456" />
19.
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
                       -->
                       <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
27.
                                cproperty name="dataSource" ref="dataSource"></property>
28.
                                <property name="configLocation" value="classpath:config/mybatis-config.xml" />
29.
                       </bean>
30.
31.
                       <!--
32.
                                3. mybatis自动扫描加载Sql映射文件/接
33.
              ☐ : MapperScannerConfigurer sqlSessionFactory
34.
                                basePackage:指定sql映射文件/接口所在的包(自动扫描)
35.
```

```
2016/5/2
          【持久化框架】SpringMVC+Spring4+Mybatis3集成,开发简单Web项目+源码下载 - 努力+坚持,而且还很年轻 - 博客频道 - CSDN.NET
    36.
              <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
    37.
                 cproperty name="basePackage" value="com.tgb.mapper"></property>
    38.
    39.
                  <property name="sqlSessionFactory" ref="sqlSessionFactory"></property>
    40.
              </bean>
    41.
              <!--
    42.
                 4. 事务管理 : DataSourceTransactionManager dataSource:引用上面定义的数据源
    43.
    44.
    45.
              <bean id="txManager"</pre>
    46.
                 class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
    47.
                 ref="dataSource" ref="dataSource">
    48
              </bean>
    49
              <!-- 5. 使用声明式事务
    50.
                  transaction-manager: 引用上面定义的事务管理器
    51.
    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 &
      <?xml version="1.0" encoding="UTF-8"?>
01.
      <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"</pre>
02.
      "http://mybatis.org/dtd/mybatis-3-config.dtd">
03.
04.
      <configuration>
05.
06.
          <!-- 实体类,简称 -设置别名 -->
07.
          <typeAliases>
                                                                   口载者
             <typeAlias alias="User" type="com.tgb.model.User" />
08.
09.
          </typeAliases>
          <!-- 实体接口映射资源 -->
10.
          <!--
11.
             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
12.
      org.mybatis.spring.mapper.MapperFactoryBean默认会去查找与xxMapper.java相同目录和名称的
      xxMapper.xml
13.
          -->
14.
          <mappers>
              <mapper resource="com/tgb/mapper/userMapper.xml" />
15.
16.
          </mappers>
17.
      </configuration>
18.
```