Maven+Spring MVC+Freemarker入门Demo

1 数据库准备

（1）建立一个名为mytest的数据库

（2）创建表

CREATE TABLE `t\_user` ( `USER\_ID` int(11) NOT NULL AUTO\_INCREMENT, `USER\_NAME` char(30) NOT NULL, `USER\_PASSWORD` char(10) NOT NULL, PRIMARY KEY (`USER\_ID`), KEY `IDX\_NAME` (`USER\_NAME`) ) ENGINE=InnoDB AUTO\_INCREMENT=11 DEFAULT CHARSET=utf8

（3）插入数据

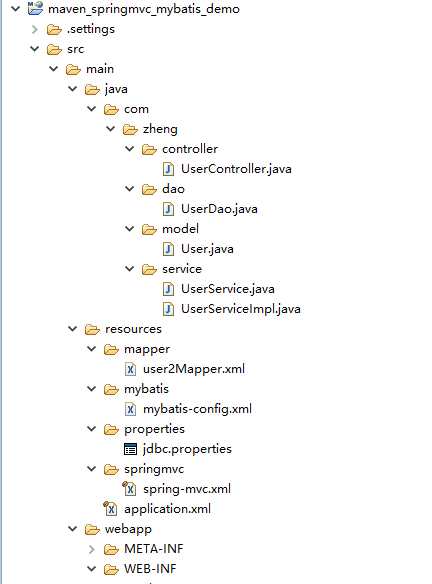
INSERT INTO t\_user (USER\_ID, USER\_NAME, USER\_PASSWORD) VALUES (1, 'Zhang San', '654321');

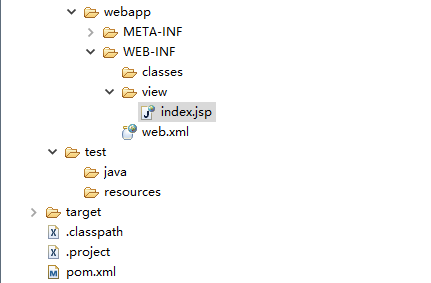
INSERT INTO t\_user (USER\_ID, USER\_NAME, USER\_PASSWORD) VALUES (2, 'Li Si', '123456');

2 建立一个名为maven\_springmvc\_mybatis\_demo的Maven Project，具体步骤见

<http://blog.csdn.net/haishu_zheng/article/details/51490299>

3 文件目录结构如下图所示





4 pom.xml的完整内容为

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"*

xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>maven\_springmvc\_mybatis\_demo</groupId>

<artifactId>maven\_springmvc\_mybatis\_demo</artifactId>

<version>0.0.1-SNAPSHOT</version>

<packaging>war</packaging>

<name>maven\_springmvc\_mybatis\_demo</name>

<description />

<properties>

<!-- spring版本号 -->

<spring.version>3.2.9.RELEASE</spring.version>

<!-- junit版本号 -->

<junit.version>4.10</junit.version>

<!-- mybatis版本号 -->

<mybatis.version>3.2.8</mybatis.version>

</properties>

<dependencies>

<!-- 添加Spring依赖 -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context-support</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aop</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-aspects</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-tx</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>${spring.version}</version>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-web</artifactId>

<version>${spring.version}</version>

</dependency>

<!--单元测试依赖 -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>${junit.version}</version>

<scope>test</scope>

</dependency>

<!--spring单元测试依赖 -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-test</artifactId>

<version>${spring.version}</version>

<scope>test</scope>

</dependency>

<!--mybatis依赖 -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>${mybatis.version}</version>

</dependency>

<!-- mybatis/spring包 -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis-spring</artifactId>

<version>1.2.2</version>

</dependency>

<!-- mysql驱动包 -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>5.1.34</version>

</dependency>

<!-- javaee-api包 注意和项目使用的JDK版本对应 -->

<dependency>

<groupId>javax</groupId>

<artifactId>javaee-api</artifactId>

<version>7.0</version>

<scope>provided</scope>

</dependency>

<!-- javaee-web-api包 注意和项目使用的JDK版本对应 -->

<dependency>

<groupId>javax</groupId>

<artifactId>javaee-web-api</artifactId>

<version>7.0</version>

<scope>provided</scope>

</dependency>

</dependencies>

</project>

5 web.xml中的完整内容为

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

<web-app xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xmlns=*"http://xmlns.jcp.org/xml/ns/javaee"* xsi:schemaLocation=*"http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"* id=*"WebApp\_ID"* version=*"3.1"*>

<display-name>maven\_springmvc\_mybatis\_demo</display-name>

<!-- 读取spring配置文件 -->

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:application.xml</param-value>

</context-param>

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<!-- springMVC核心配置 -->

<servlet>

<servlet-name>dispatcherServlet</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<!--spingMVC的配置路径 -->

<param-value>classpath:springmvc/spring-mvc.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<!-- 拦截设置 -->

<servlet-mapping>

<servlet-name>dispatcherServlet</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

6 User.java的完整代码为

**package** com.zheng.model;

**public** **class** User {

**private** Integer userId;

**private** String userName;

**private** String userPassword;

**public** Integer getUserId() {

**return** userId;

}

**public** **void** setUserId(Integer userId) {

**this**.userId = userId;

}

**public** String getUserName() {

**return** userName;

}

**public** **void** setUserName(String userName) {

**this**.userName = userName;

}

**public** String getUserPassword() {

**return** userPassword;

}

**public** **void** setUserPassword(String userPassword) {

**this**.userPassword = userPassword;

}

}

7 UserDao.java的完整代码为

**package** com.zheng.dao;

**import** com.zheng.model.User;

**public** **interface** UserDao {

**public** User selectUserById(Integer userId);

}

8 UserService.java的完整代码为

**package** com.zheng.service;

**import** com.zheng.model.User;

**public** **interface** UserService {

User selectUserById(Integer userId);

}

9 UserServiceImpl.java的完整代码为

package com.zheng.service;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import com.zheng.dao.UserDao;

import com.zheng.model.User;

import com.zheng.service.UserService;

@Service

public class UserServiceImpl implements UserService {

@Autowired

private UserDao userDao;

public User selectUserById(Integer userId) {

return userDao.selectUserById(userId);

}

}

10 UserController.java的完整代码为

package com.zheng.controller;

import javax.annotation.Resource;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.servlet.ModelAndView;

import com.zheng.model.User;

import com.zheng.service.UserService;

@Controller

public class UserController {

@Resource

private UserService userService;

@RequestMapping("/showUserInfo")

public ModelAndView getIndex(){

ModelAndView mav = new ModelAndView("index");

User user = userService.selectUserById(1);

mav.addObject("user", user);

return mav;

}

}

11 User2Mapper.xml的完整代码为

<?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.zheng.dao.UserDao"*>

<!--设置model类和数据库中表的字段一一对应，注意数据库字段和model类中的字段名称不致，此处一定要！-->

<resultMap id=*"BaseResultMap"* type=*"com.zheng.model.User"*>

<id column=*"USER\_ID"* property=*"userId"* jdbcType=*"INTEGER"* />

<result column=*"USER\_NAME"* property=*"userName"* jdbcType=*"CHAR"* />

<result column=*"USER\_PASSWORD"* property=*"userPassword"* jdbcType=*"CHAR"* />

</resultMap>

<!-- 查询单条记录 -->

<select id=*"selectUserById"* parameterType=*"int"* resultMap=*"BaseResultMap"*>

SELECT \* FROM t\_user WHERE USER\_ID = #{userId}

</select>

</mapper>

12 mybatis-config.xml的完整代码为

<?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">

<configuration>

</configuration>

13 jdbc.properties的完整代码为

jdbc\_driverClassName=com.mysql.jdbc.Driver

jdbc\_url=jdbc:mysql://localhost:3306/mytest

jdbc\_username=root

jdbc\_password=root

14 spring-mvc.xml的完整代码为

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

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

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

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

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

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

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.2.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.2.xsd*

*http://www.springframework.org/schema/mvc*

*http://www.springframework.org/schema/mvc/spring-mvc-3.2.xsd"*>

<mvc:annotation-driven />

<!-- 扫描controller（controller层注入） -->

<context:component-scan base-package=*"com.zheng.controller"*/>

<!-- 对模型视图添加前后缀 -->

<bean id=*"viewResolver"* class=*"org.springframework.web.servlet.view.InternalResourceViewResolver"*

p:prefix=*"/WEB-INF/view/"* p:suffix=*".jsp"*/>

</beans>

15 application.xml的完整代码为

<?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"*

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

xsi:schemaLocation=*"*

*http://www.springframework.org/schema/beans*

*http://www.springframework.org/schema/beans/spring-beans-3.0.xsd*

*http://www.springframework.org/schema/aop*

*http://www.springframework.org/schema/aop/spring-aop-3.0.xsd*

*http://www.springframework.org/schema/context*

*http://www.springframework.org/schema/context/spring-context-3.0.xsd"*>

<!-- 引入jdbc配置文件 -->

<bean id=*"propertyConfigurer"* class=*"org.springframework.beans.factory.config.PropertyPlaceholderConfigurer"*>

<property name=*"locations"*>

<list>

<value>classpath:properties/\*.properties</value>

<!--要是有多个配置文件，只需在这里继续添加即可 -->

</list>

</property>

</bean>

<!-- 配置数据源 -->

<bean id=*"dataSource"*

class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<!-- 不使用properties来配置 -->

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

<property name="url" value="jdbc:mysql://localhost:3306/learning" />

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

<property name="password" value="christmas258@" /> -->

<!-- 使用properties来配置 -->

<property name=*"driverClassName"*>

<value>${jdbc\_driverClassName}</value>

</property>

<property name=*"url"*>

<value>${jdbc\_url}</value>

</property>

<property name=*"username"*>

<value>${jdbc\_username}</value>

</property>

<property name=*"password"*>

<value>${jdbc\_password}</value>

</property>

</bean>

<!-- 自动扫描了所有的XxxxMapper.xml对应的mapper接口文件，这样就不用一个一个手动配置Mpper的映射了，只要Mapper接口类和Mapper映射文件对应起来就可以了。 -->

<bean class=*"org.mybatis.spring.mapper.MapperScannerConfigurer"*>

<property name=*"basePackage"*

value=*"com.zheng.dao"* />

</bean>

<!-- 配置Mybatis的文件 ，mapperLocations配置\*\*Mapper.xml文件位置，configLocation配置mybatis-config文件位置-->

<bean id=*"sqlSessionFactory"* class=*"org.mybatis.spring.SqlSessionFactoryBean"*>

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

<property name=*"mapperLocations"* value=*"classpath:mapper/\*.xml"*/>

<property name=*"configLocation"* value=*"classpath:mybatis/mybatis-config.xml"* />

<!-- <property name="typeAliasesPackage" value="com.tiantian.ckeditor.model"

/> -->

</bean>

<!-- 自动扫描注解的bean -->

<context:component-scan base-package=*"com.zheng.service"* />

</beans>

16 index.jsp中的完整代码为：

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"*%>

<html>

<body>

<h2>User Info</h2>

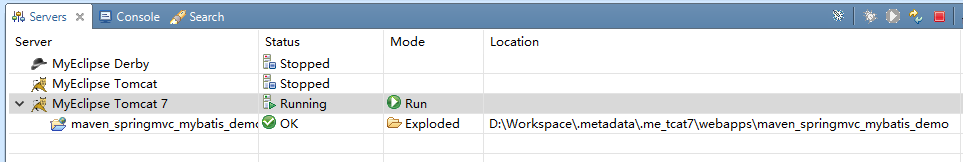
用户名： ${user.userName}<br>

密码： ${user.userPassword}<br>

</body>

</html>

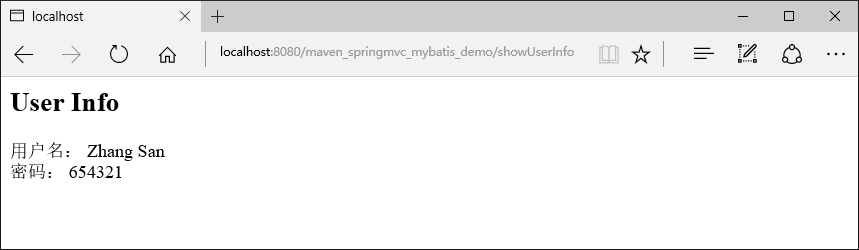
17 将项目添加进Tomcat 7中并运行



在浏览器中输入

<http://localhost:8080/maven_springmvc_mybatis_demo/showUserInfo>

显示结果为



18 源码下载地址

CSDN：

Github：