介绍一下SpringBoot整合mybatis，数据库选用的是mysql。

首先创建数据库

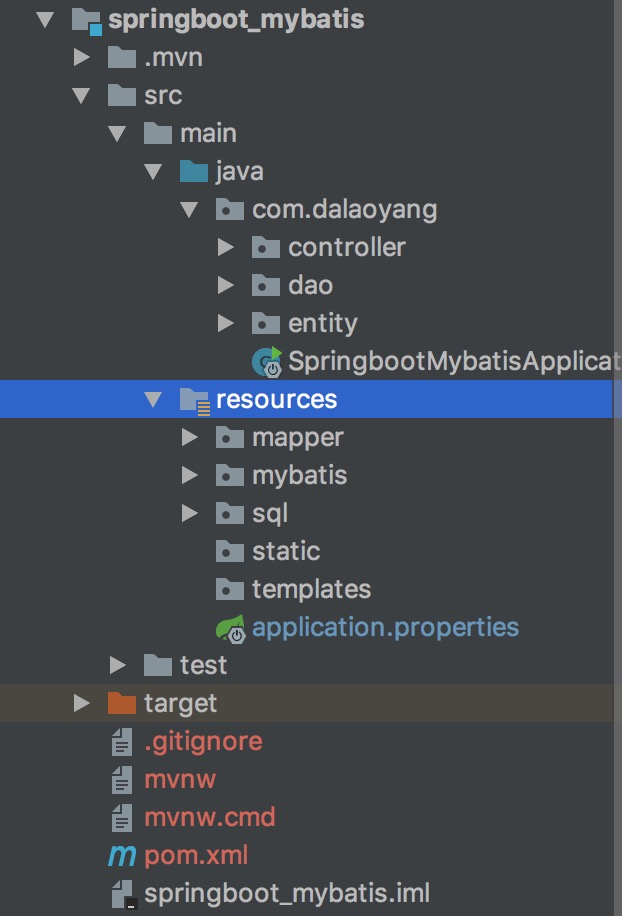
CREATE DATABASE test;

建表以及插入初始数据(sql是从navicat中导出的)

SET NAMES utf8mb4; SET FOREIGN\_KEY\_CHECKS = 0; -- ---------------------------- -- Table structure for user -- ---------------------------- DROP TABLE IF EXISTS `user`; CREATE TABLE `user` ( `id` int(11) NOT NULL AUTO\_INCREMENT, `user\_name` varchar(255) NOT NULL, `user\_password` varchar(255) DEFAULT NULL, PRIMARY KEY (`id`) ) ENGINE=InnoDB AUTO\_INCREMENT=13 DEFAULT CHARSET=latin1; -- ---------------------------- -- Records of user -- ---------------------------- BEGIN; INSERT INTO `user` VALUES (1, 'dalaoyang', '13'); INSERT INTO `user` VALUES (2, 'xiaoli', '123'); INSERT INTO `user` VALUES (3, 'xiaoxiongmao', '123'); COMMIT; SET FOREIGN\_KEY\_CHECKS = 1;

下图为项目目录结构，  
java—  
controller包负责测试整合  
dao包作为数据操作层  
entity作为数据实体类

resources—  
mapper写dao层对应实现的sql  
mybatis里面是mybatis配置，包含typeAlias等等  
sql里面放的是上面写的建表数据及sql



接下来直接上代码，启动类没有修改，代码如下

package com.dalaoyang; import org.mybatis.spring.annotation.MapperScan; import org.springframework.boot.SpringApplication; import org.springframework.boot.autoconfigure.EnableAutoConfiguration; import org.springframework.boot.autoconfigure.SpringBootApplication; import org.springframework.boot.autoconfigure.jdbc.DataSourceAutoConfiguration; @SpringBootApplication public class SpringbootMybatisApplication { public static void main(String[] args) { SpringApplication.run(SpringbootMybatisApplication.class, args);

}

}

application.properties包含了数据库配置，mybatis配置，代码如下:

##端口号 server.port=8888 ##检查 mybatis 配置是否存在，一般命名为 mybatis-config.xml mybatis.check-config-location =true ##配置文件位置 mybatis.config-location=classpath:mybatis/mybatis-config.xml ## mapper xml 文件地址 mybatis.mapper-locations=classpath\*:mapper/\*Mapper.xml ##日志级别 logging.level.com.yang.dao=debug ##数据库url spring.datasource.url=jdbc:mysql://localhost:3306/test?characterEncoding=utf8&useSSL=false ##数据库用户名 spring.datasource.username=root ##数据库密码 spring.datasource.password=root ##数据库驱动 spring.datasource.driver-class-name=com.mysql.jdbc.Driver

实体类User

package com.dalaoyang.entity; import org.apache.ibatis.type.Alias; /\*\* \* @author dalaoyang \* @Description \* @project springboot\_learn \* @package com.dalaoyang.entity \* @email 397600342@qq.com \* @date 2018/4/5 \*/ @Alias("user") public class User { private int id; private String user\_name; private String user\_password; public User(String user\_name, String user\_password) { this.user\_name = user\_name; this.user\_password = user\_password; } public User(int id, String user\_name, String user\_password) { this.id = id; this.user\_name = user\_name; this.user\_password = user\_password; } public int getId() { return id; } public void setId(int id) { this.id = id; } public String getUser\_name() { return user\_name; } public void setUser\_name(String user\_name) { this.user\_name = user\_name; } public String getUser\_password() { return user\_password; } public void setUser\_password(String user\_password) { this.user\_password = user\_password; } }

dao层代码

package com.dalaoyang.dao; import com.dalaoyang.entity.User; import org.apache.ibatis.annotations.Mapper; import java.util.List; /\*\* \* @author dalaoyang \* @Description \* @project springboot\_learn \* @package com.dalaoyang.dao \* @email 397600342@qq.com \* @date 2018/4/5 \*/ @Mapper public interface UserMapper { User findUserByUsername(String username); void updateUserByUsername(User user); void deleteUserByUsername(String username); void saveUser(User user); List<User> getUserList(); }

UserMapper.xml代码

<?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.dalaoyang.dao.UserMapper"> <resultMap id="user" type="com.dalaoyang.entity.User"/> <parameterMap id="user" type="com.dalaoyang.entity.User"/> <select id="findUserByUsername" parameterType="String" resultMap="user"> SELECT \* FROM user WHERE user\_name=#{1} </select> <update id="updateUserByUsername" parameterMap="user"> UPDATE USER SET USER\_PASSWORD=#{user\_password} WHERE USER\_NAME=#{user\_name} </update> <delete id="deleteUserByUsername" parameterType="String"> DELETE FROM USER WHERE USER\_NAME=#{1} </delete> <!-- 使用alias自定义的parameterType--> <insert id="saveUser" parameterType="user"> INSERT INTO USER (user\_password,user\_name) VALUES (#{user\_password},#{user\_name}) </insert> <select id="getUserList" resultMap="user"> SELECT \* FROM USER </select> </mapper>

mybatis-config.xml

<?xml version="1.0" encoding="UTF-8"?> <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD SQL Map Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd"> <configuration> <typeAliases> <typeAlias alias="Integer" type="java.lang.Integer" /> <typeAlias alias="Long" type="java.lang.Long" /> <typeAlias alias="HashMap" type="java.util.HashMap" /> <typeAlias alias="LinkedHashMap" type="java.util.LinkedHashMap" /> <typeAlias alias="ArrayList" type="java.util.ArrayList" /> <typeAlias alias="LinkedList" type="java.util.LinkedList" /> <typeAlias alias="user" type="com.dalaoyang.entity.User"/> </typeAliases> </configuration>

pom文件

<?xml version="1.0" encoding="UTF-8"?> <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>com.dalaoyang</groupId> <artifactId>springboot\_mybatis</artifactId> <version>0.0.1-SNAPSHOT</version> <packaging>jar</packaging> <name>springboot\_mybatis</name> <description>springboot\_mybatis</description> <parent> <groupId>org.springframework.boot</groupId> <artifactId>spring-boot-starter-parent</artifactId> <version>1.5.9.RELEASE</version> <relativePath/> <!-- lookup parent from repository --> </parent> <properties> <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding> <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding> <java.version>1.8</java.version> </properties> <dependencies> <dependency> <groupId>org.springframework.boot</groupId> <artifactId>spring-boot-starter-web</artifactId> </dependency> <dependency> <groupId>org.mybatis.spring.boot</groupId> <artifactId>mybatis-spring-boot-starter</artifactId> <version>1.3.1</version> </dependency> <dependency> <groupId>org.springframework.boot</groupId> <artifactId>spring-boot-devtools</artifactId> <scope>runtime</scope> </dependency> <dependency> <groupId>mysql</groupId> <artifactId>mysql-connector-java</artifactId> <scope>runtime</scope> </dependency> <dependency> <groupId>org.springframework.boot</groupId> <artifactId>spring-boot-starter-test</artifactId> <scope>test</scope> </dependency> </dependencies> <build> <plugins> <plugin> <groupId>org.springframework.boot</groupId> <artifactId>spring-boot-maven-plugin</artifactId> </plugin> </plugins> </build> </project>

UserController

package com.dalaoyang.controller; import com.dalaoyang.dao.UserMapper; import com.dalaoyang.entity.User; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.web.bind.annotation.RequestMapping; import org.springframework.web.bind.annotation.RestController; import java.util.List; /\*\* \* @author dalaoyang \* @Description \* @project springboot\_learn \* @package com.dalaoyang.controller \* @email 397600342@qq.com \* @date 2018/4/5 \*/ @RestController public class UserController { @Autowired UserMapper userMapper; //http://localhost:8888/getUser?username=xiaoli2 @RequestMapping("/getUser") public String getUser(String username){ User user =userMapper.findUserByUsername(username); return user!=null ? username+"的密码是："+user.getUser\_password():"不存在用户名为"+username+"的用户"; } //http://localhost:8888/updateUser?username=xiaoli2&password=123 @RequestMapping("/updateUser") public String updateUser(String password,String username){ User user = new User(username,password); userMapper.updateUserByUsername(user); return "success!"; } //http://localhost:8888/addUser?username=xiaoli2&password=123 @RequestMapping("/addUser") public String addUser(String username,String password){ User user = new User(username,password); userMapper.saveUser(user); return "success!"; } //http://localhost:8888/addUser?username=xiaoli2 @RequestMapping("/deleteUser") public String deleteUser(String username){ userMapper.deleteUserByUsername(username); return "success!"; } //http://localhost:8888/getUserList @RequestMapping("/getUserList") public List getUserList(String username, String password){ return userMapper.getUserList(); } }

启动项目，访问controller上面对应的注释上的地址即可以测试，  
其中包含了简单的增删改查，SpringBoot整合Mybatis就这样完成了。