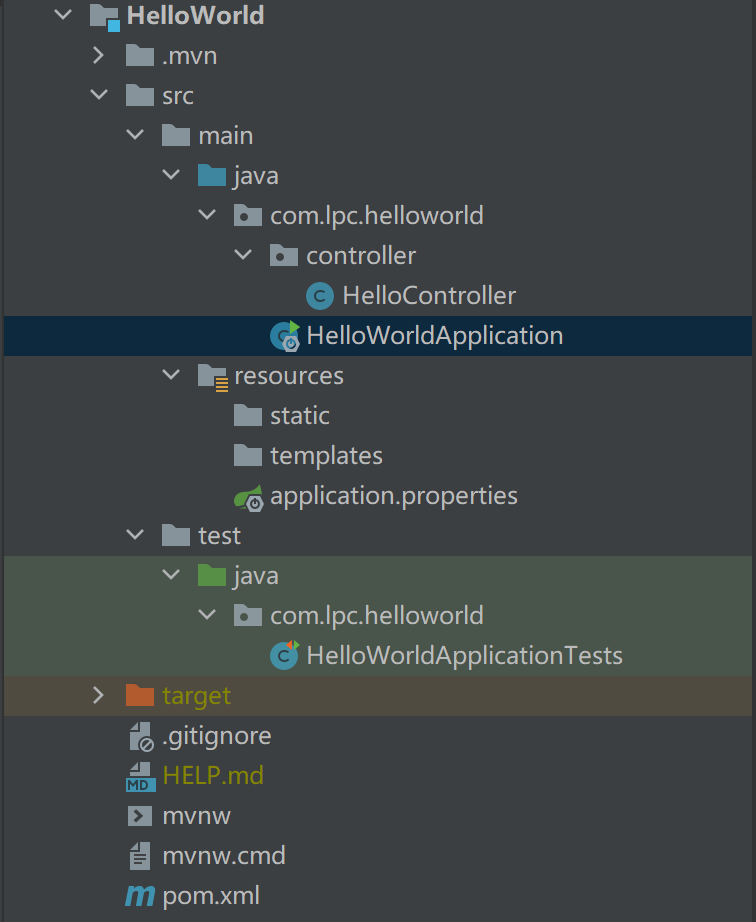
# 一、Hello World

### 1、目录结构



### 2、代码

* HelloController代码

package com.lpc.helloworld.controller;  
  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;  
  
/\*\*  
 \* @author byu\_rself  
 \* @create 2021/9/7 21:58  
 \*/  
  
@RestController  
public class HelloController {  
  
 @GetMapping("/hello")  
 public String hello(){  
 return "Hello World!";  
 }  
  
}

* HelloWorldApplication代码

package com.lpc.helloworld;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class HelloWorldApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.run(HelloWorldApplication.class, args);  
 }  
  
}

### 3、运行效果

##### >浏览器内运行



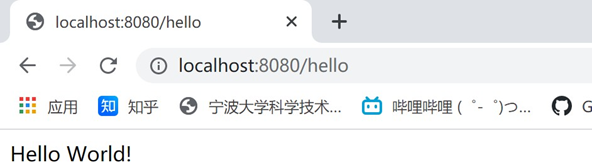
##### >cmd命令行中运行

对代码打包



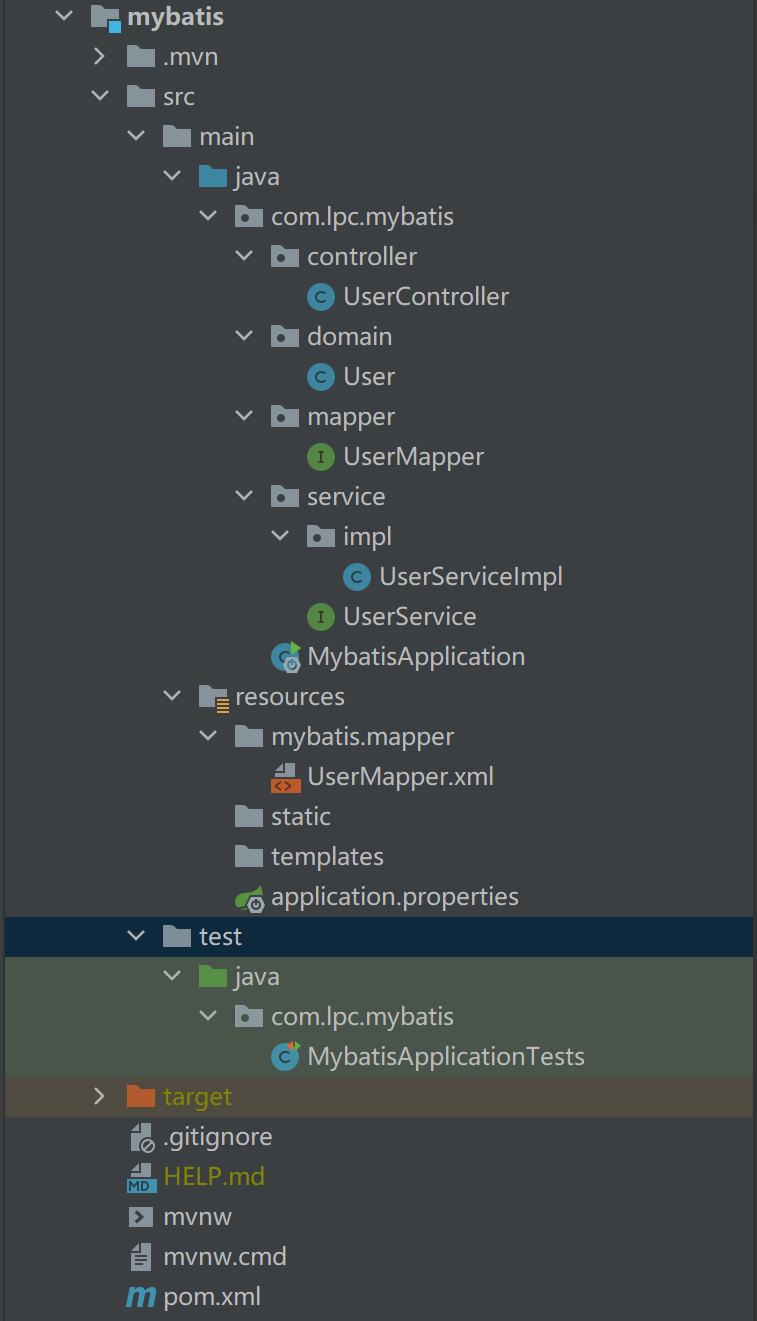
在cmd中输入指令



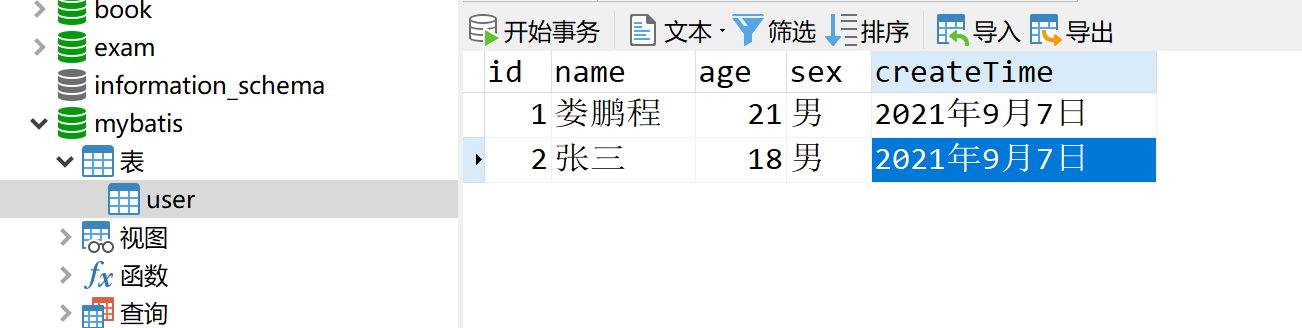


# 二、Mybatis

### 1、目录结构



### 2、初始数据库



### 3、代码

* User代码
* package com.lpc.mybatis.domain;  
    
  import lombok.AllArgsConstructor;  
  import lombok.Data;  
  import lombok.NoArgsConstructor;  
    
  /\*\*  
   \* @author byu\_rself  
   \* @create 2021/9/7 20:46  
   \*/  
    
  @Data  
  @NoArgsConstructor  
  @AllArgsConstructor  
  public class User {  
    
   private Integer id;  
   private String name;  
   private Integer age;  
   private String sex;  
   private String createTime;  
     
  }
* application.properties代码
* spring.datasource.username=root  
  spring.datasource.password=123456  
  spring.datasource.url=jdbc:mysql://localhost:3306/mybatis?serverTimezone=UTC&useUnicode=true&characterEncoding=utf-8  
  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
    
  # 整合mybatis  
  mybatis.type-aliases-package=com.lpc.mybatis.domain  
  mybatis.mapper-locations=classpath:mybatis/mapper/\*.xml
* 在test包中测试mybatis是否配置成功
* package com.lpc.mybatis;  
    
  import org.junit.jupiter.api.Test;  
  import org.springframework.beans.factory.annotation.Autowired;  
  import org.springframework.boot.test.context.SpringBootTest;  
    
  import javax.sql.DataSource;  
  import java.sql.SQLException;  
    
  @SpringBootTest  
  class MybatisApplicationTests {  
    
   @Autowired  
   DataSource dataSource;  
    
   @Test  
   void contextLoads() throws SQLException {  
   System.out.println(dataSource.getClass());  
   System.out.println(dataSource.getConnection());  
   }  
    
  }
* UserMapper代码
* package com.lpc.mybatis.mapper;  
    
  import com.lpc.mybatis.domain.User;  
  import org.apache.ibatis.annotations.Mapper;  
  import org.springframework.stereotype.Repository;  
    
  import java.util.List;  
    
  /\*\*  
   \* @author byu\_rself  
   \* @create 2021/9/7 20:50  
   \*/  
    
  @Mapper  
  @Repository  
  public interface UserMapper {  
    
   List<User> selectAllUser();  
    
   User queryUserById(Integer id);  
    
   int addUser(User user);  
    
   int updateUserById(User user);  
    
   int deleteUserById(int id);  
    
  }
* 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.lpc.mybatis.mapper.UserMapper">  
    
   <select id="selectAllUser" resultType="User">  
   select \* from user  
   </select>  
    
   <select id="queryUserById" resultType="User">  
   select \* from user where id = #{id}  
   </select>  
    
   <insert id="addUser" parameterType="User">  
   insert into user (id,name,age,sex,createTime) values (#{id},#{name},#{age},#{sex},#{createTime})  
   </insert>  
    
   <update id="updateUserById" parameterType="User">  
   update user set name = #{name},age = #{age},sex=#{sex},createTime=#{createTime} where id = #{id}  
   </update>  
    
   <delete id="deleteUserById" parameterType="Integer">  
   delete from user where id = #{id}  
   </delete>  
    
  </mapper>
* UserService代码
* package com.lpc.mybatis.service;  
    
  import com.lpc.mybatis.domain.User;  
    
  import java.util.List;  
    
  /\*\*  
   \* @author byu\_rself  
   \* @create 2021/9/7 21:44  
   \*/  
    
  public interface UserService {  
    
   List<User> selectAllUser();  
    
   User queryUserById(Integer id);  
    
   int addUser(User user);  
    
   int updateUserById(User user);  
    
   int deleteUserById(int id);  
    
  }
* UserServiceImpl代码
* package com.lpc.mybatis.service.impl;  
    
  import com.lpc.mybatis.domain.User;  
  import com.lpc.mybatis.mapper.UserMapper;  
  import com.lpc.mybatis.service.UserService;  
  import org.springframework.beans.factory.annotation.Autowired;  
  import org.springframework.stereotype.Service;  
    
  import java.util.List;  
    
  /\*\*  
   \* @author byu\_rself  
   \* @create 2021/9/7 21:46  
   \*/  
  @Service  
  public class UserServiceImpl implements UserService {  
    
   @Autowired  
   private UserMapper UserMapper;  
    
   @Override  
   public List<User> selectAllUser() {  
   List<User> userList = UserMapper.selectAllUser();  
   return userList;  
   }  
    
   @Override  
   public User queryUserById(Integer id) {  
   return UserMapper.queryUserById(id);  
   }  
    
   @Override  
   public int addUser(User user) {  
   return UserMapper.addUser(user);  
   }  
    
   @Override  
   public int updateUserById(User user) {  
   return UserMapper.updateUserById(user);  
   }  
    
   @Override  
   public int deleteUserById(int id) {  
   return UserMapper.deleteUserById(id);  
   }  
  }
* UserController代码
* package com.lpc.mybatis.controller;  
    
  import com.lpc.mybatis.domain.User;  
  import com.lpc.mybatis.service.UserService;  
  import org.springframework.beans.factory.annotation.Autowired;  
  import org.springframework.web.bind.annotation.GetMapping;  
  import org.springframework.web.bind.annotation.RestController;  
    
  import java.util.List;  
    
  /\*\*  
   \* @author byu\_rself  
   \* @create 2021/9/7 21:09  
   \*/  
    
  @RestController  
  public class UserController {  
    
   @Autowired  
   private UserService userService;  
    
   @GetMapping("/selectAllUser")  
   public List<User> selectAllUser(){  
   List<User> userList = userService.selectAllUser();  
   for (User user : userList) {  
   System.out.println(user);  
   }  
   return userList;  
   }  
    
   @GetMapping("/queryUserById")  
   public User queryUserById(){  
   User user = userService.queryUserById(1);  
   return user;  
   }  
    
   @GetMapping("/addUser")  
   public String addUser(){  
   userService.addUser(new User(null,"张三",18,"男","2021年9月7日"));  
   return "添加成功";  
   }  
    
   @GetMapping("/updateUser")  
   public String updateUser(){  
   userService.updateUserById(new User(1,"Lpc",18,"男","2021年9月9日"));  
   return "修改成功";  
   }  
    
   @GetMapping("/deleteUserById")  
   public String deleteUserById(){  
   userService.deleteUserById(2);  
   return "删除成功";  
   }  
    
  }

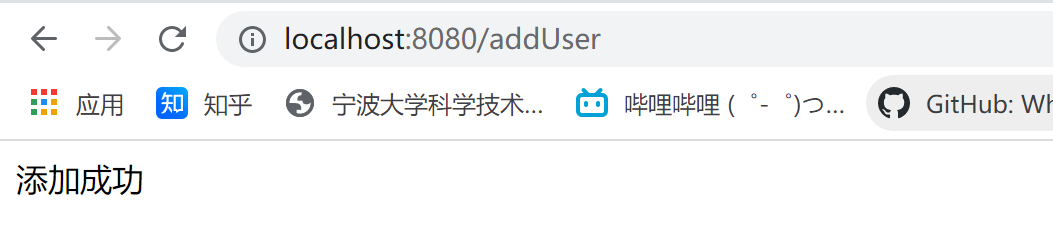
### 4、运行效果

##### >查询所有用户

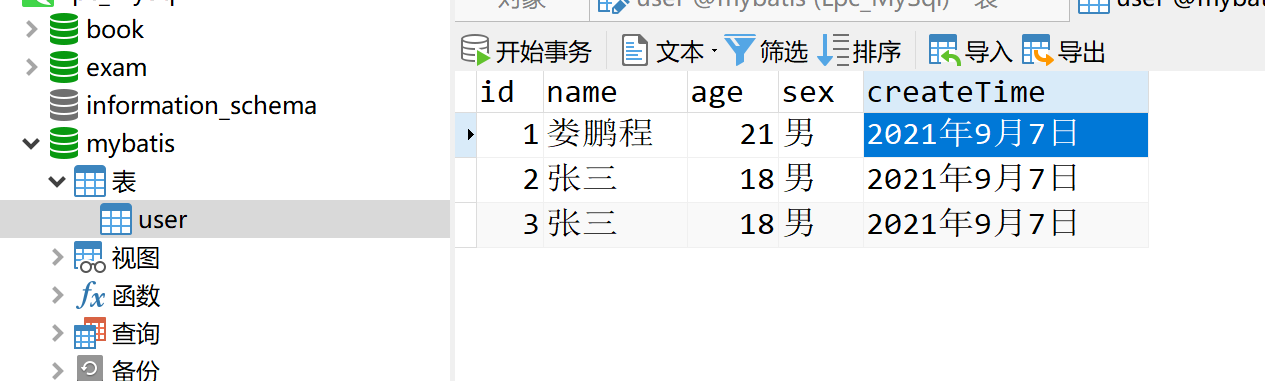
##### fig:>根据id查询用户（此时id=1)



##### >添加用户



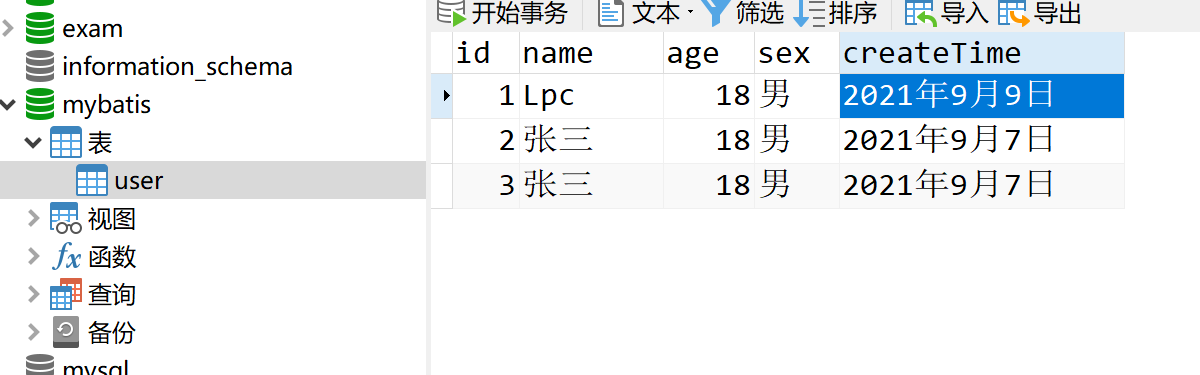
此时数据库中新增一条记录



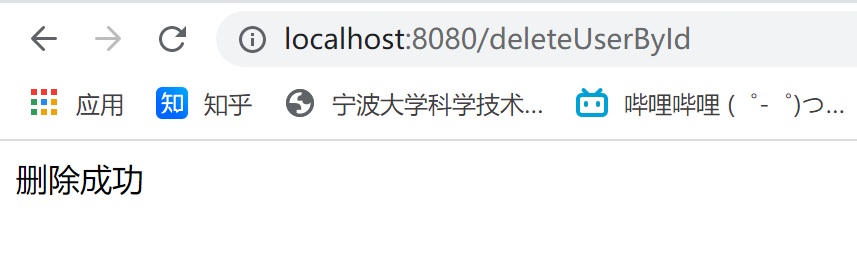
##### >修改用户



此时数据库第一条记录被修改



##### >删除用户



此时数据库第二条记录被删除

