# Spring boot+MyBatis+PageHelper与前端交互JSON数据

## 1、pom.xml

|  |
| --- |
| <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>org.leo</groupId>  <artifactId>springboot</artifactId>  <version>0.0.1-SNAPSHOT</version>  <packaging>jar</packaging>  <name>springboot</name>  <url>http://maven.apache.org</url>  <!-- Inherit defaults from Spring Boot -->  <!-- https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-parent -->  <parent>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-parent</artifactId>  <version>1.5.4.RELEASE</version>  </parent>  <properties>  <java.version>1.8</java.version>  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>  </properties>  <dependencies>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  <dependency>  <groupId>com.alibaba</groupId>  <artifactId>druid</artifactId>  <version>1.0.31</version>  </dependency>  <dependency>  <groupId>org.mybatis.spring.boot</groupId>  <artifactId>mybatis-spring-boot-starter</artifactId>  <version>1.3.0</version>  </dependency>  <!-- https://mvnrepository.com/artifact/com.github.pagehelper/pagehelper-spring-boot-starter -->  <dependency>  <groupId>com.github.pagehelper</groupId>  <artifactId>pagehelper-spring-boot-starter</artifactId>  <version>1.1.2</version>  </dependency>  <!-- mysql -->  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-java</artifactId>  </dependency>  </dependencies>    <build>  <plugins>  <plugin>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-maven-plugin</artifactId>  <dependencies>  <dependency>  <groupId>org.springframework</groupId>  <artifactId>springloaded</artifactId>  <version>1.2.7.RELEASE</version>  </dependency>  </dependencies>  </plugin>  </plugins>  </build>  </project> |

## 2、application.yml

文件放在src\main\resources\config目录下

|  |
| --- |
| spring:  aop:  proxy-target-class: true  datasource:  name: ssm  url: jdbc:mysql://192.168.56.102:3306/ssm ?useUnicode=true&characterEncoding=utf-8&useSSL=false  username: root  password: MyNewPwd4!  # 使用druid数据源  type: com.alibaba.druid.pool.DruidDataSource  driver-class-name: com.mysql.jdbc.Driver  filters: stat  maxActive: 20  initialSize: 1  maxWait: 60000  minIdle: 1  timeBetweenEvictionRunsMillis: 60000  minEvictableIdleTimeMillis: 300000  validationQuery: select 'x'  testWhileIdle: true  testOnBorrow: false  testOnReturn: false  poolPreparedStatements: true  maxOpenPreparedStatements: 20  mybatis:  mapperLocations: classpath:mapper/\*.xml  typeAliasesPackage: org.leo.springboot.pojo  #pagehelper.  pagehelper:  autoDialect: true  closeConn: true |

## 3、mapper

mapper的XML放在src\main\resources\mapper下

|  |
| --- |
| <?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=*"org.leo.springboot.mapper.TUserMapper"*>  <select id=*"findList"* resultType=*"org.leo.springboot.pojo.TUser"*  parameterType=*"org.leo.springboot.pojo.TUser"*>  SELECT id, username, gender, intime FROM t\_user  <where>  <if test=*"id != null"*>  and id=#{id}  </if>  <if test=*"username != null"*>  and username=#{username}  </if>  <if test=*"gender != null"*>  and gender=#{gender}  </if>  </where>  </select>  </mapper> |

Mapper接口：

|  |
| --- |
| package org.leo.springboot.mapper;  import java.util.List;  import org.apache.ibatis.annotations.Mapper;  import org.leo.springboot.pojo.TUser;  @Mapper  public interface TUserMapper {  public List<TUser> findList(TUser user);  } |

## 4、Service

接口就不附上了。

|  |
| --- |
| package org.leo.springboot.service.impl;  import java.util.List;  import org.leo.springboot.mapper.TUserMapper;  import org.leo.springboot.pojo.TUser;  import org.leo.springboot.service.IUserService;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import org.springframework.transaction.annotation.Transactional;  @Service  @Transactional  public class UserService implements IUserService {  @Autowired  private TUserMapper tUserMapper;  @Override  public List<TUser> findList(TUser user) {  return tUserMapper.findList(user);  }  } |

## 5、Controller

|  |
| --- |
| package org.leo.springboot.controller;  import java.util.List;  import org.leo.springboot.pojo.TUser;  import org.leo.springboot.service.IUserService;  import org.leo.springboot.vo.UserReq;  import org.leo.springboot.vo.UserRes;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.boot.autoconfigure.EnableAutoConfiguration;  import org.springframework.web.bind.annotation.PathVariable;  import org.springframework.web.bind.annotation.RequestBody;  import org.springframework.web.bind.annotation.RequestMapping;  import org.springframework.web.bind.annotation.RequestMethod;  import org.springframework.web.bind.annotation.ResponseBody;  import org.springframework.web.bind.annotation.RestController;  import com.github.pagehelper.PageHelper;  @RestController  @EnableAutoConfiguration  @RequestMapping("/user")  public class UserController {  @Autowired  private IUserService userService;  @RequestMapping("/{id}")  List<TUser> testGet(@PathVariable("id") int id) {  List<TUser> users1 = userService.findList(new TUser(id, null, null, null));  System.out.println("---" + users1.toString());  PageHelper.startPage(1, 1);  List<TUser> users2 = userService.findList(new TUser());  System.out.println("===" + users2.toString());  List<TUser> users3 = userService.findList(new TUser());  System.out.println("用户id是 " + id + " 的username是 " + users1.get(0).getUsername() + " 。员工总数是" + users3.size());  return users3;  }  @RequestMapping(value = "/testpost", method = RequestMethod.POST)  @ResponseBody  UserRes testPost(@RequestBody UserReq req) {  System.out.println(req.toString());  List<TUser> users1 = userService.findList(new TUser(req.getId(), null, null, null));  System.out.println(users1.toString());  return new UserRes(0, "成功", users1.get(0));  }  } |

这里有两个方法，都返回json数据：

1. testGet，用来演示get方法，其中包含使用PageHelper的代码，用来分页。
2. testPost，用来演示post接收json数据。

里面具体的实体类就不附上了。

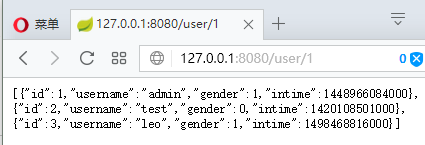
## 6、Spring boot启动

|  |
| --- |
| package org.leo.springboot;  import org.mybatis.spring.annotation.MapperScan;  import org.springframework.boot.SpringApplication;  import org.springframework.boot.autoconfigure.SpringBootApplication;  @SpringBootApplication  public class Application {  public static void main(String[] args) {  SpringApplication.run(Application.class);  }  } |

直接运行。

## 7、测试

GET：



POST：

