

**Java training**

**实训报告书**

专业： 网络工程（NIIT项目）

班级： 2016030203

学号： 201603020325

姓名： 唐永海

年 月 日

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Major  专业 | 网络工程(NIIT) | | Course name课程名称 | | Java training | | |
| Guide Teacher 指导老师 | 廖昌旺 | Place 实验地点 | | 11207 | | Date 实验日期 | 2023年9 月12 日 |
| Student Name 学生姓名 | 唐永海 | Batch 班　级 | | Java | | Student Id 学　号 | 201603020325 |
| Activity  &  Exercises 活动与练习 | Sprint02-用户管理模块实训报告：Practice1批量删除和Practice2 分页查询用户信息 | | | | | | |
| Details for  Exercise  练习相关细节 | Exercise练习: Exercise1 | | | | | | |
| Request要求:  (Problem Statement)关于问题的描述  Practice1批量删除 | | | | | | |
| Solution解决:  1.type 服务端修改 ConsumerController.java 前端传递的多个参数名相同，后端使用数组接收  */\*\*  \* 批量删除多个用户  \*  \* @param id  \* @return  \*/* @DeleteMapping("user/deleteIds") public R deleteUsers(@RequestParam String[] id) {  return consumerService.deleteUsers(id); }  2.type 服务端修改 ConsumerServiceImpl.java ConsumerService.java增加接口  */\*\*  \* 批量删除用户接口  \* @param ids  \* @return  \*/* R deleteUsers(String[] ids);  */\*\*  \* 批量删除多个用户  \*/* @Override public R deleteUsers(String[] ids) {  *//把数组转成字符串  //把数组中参数组织成SQL语句需要的格式* StringBuffer sb = new StringBuffer();  for (String i : ids) {  sb.append(i + ",");  }  *//截掉最后一个逗号* sb.deleteCharAt(sb.length() - 1);  if (consumerMapper.deleteByIds(sb.toString()) > 0) {  return R.*success*("删除成功");  } else {  return R.*error*("删除失败");  } }  3.type 服务端修改 ConsumerMapper.java  *//添加自定义方法，用来完成批量删除  /\*\*  \* 批量删除用户  \* @param ids  \* @return  \*/* int deleteByIds(String ids);  4.type . 服务端修改 ConsumerMapper.xml  *<!--根据id批量删除用户-->* <delete id="deleteByIds">  delete from consumer where id in (${ids})  </delete>  5.type启动前端，后端 | | | | | | |
| Source code源代码:  ConsumerController.java  package com.example.onlinemusicservice.controller;  import com.baomidou.mybatisplus.core.metadata.IPage; import com.baomidou.mybatisplus.extension.plugins.pagination.Page; import com.example.onlinemusicservice.common.R; import com.example.onlinemusicservice.model.domain.Consumer; import com.example.onlinemusicservice.service.ConsumerService; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.web.bind.annotation.DeleteMapping; import org.springframework.web.bind.annotation.GetMapping; import org.springframework.web.bind.annotation.RequestParam; import org.springframework.web.bind.annotation.RestController;  */\*\*  \* admin-用户管理控制类（controller层）  \*/* @RestController public class ConsumerController {  @Autowired  private ConsumerService consumerService;   */\*\*  \* 管理界面调用：返回所有用户进行分页管理  \* @param curPage 当前页  \* @param pageSize 页的大小  \* @return  \*/* @GetMapping("/user")  public R allUser(@RequestParam(defaultValue = "1") int curPage, @RequestParam(defaultValue = "1") int pageSize) {  *//分* IPage<Consumer> page = new Page<>();  *//设置当前页* page.setCurrent(curPage);  *//设置每页显示数据条数* page.setSize(pageSize);  return consumerService.allUser(page);  }   */\*\*  \* 删除用户（根据id删除）  \*/* @GetMapping("/user/delete")  public R deleteUser(@RequestParam int id) {  return consumerService.deleteUser(id);  }   */\*\*  \* 批量删除多个用户  \*  \* @param id  \* @return  \*/* @DeleteMapping("user/deleteIds")  public R deleteUsers(@RequestParam String[] id) {  return consumerService.deleteUsers(id);  } }  ConsumerMapper:  package com.example.onlinemusicservice.mapper;  import com.baomidou.mybatisplus.core.mapper.BaseMapper; import com.example.onlinemusicservice.model.domain.Consumer;  */\*\*  \* admin-用户管理（Mapper层接口）  \*/* public interface ConsumerMapper extends BaseMapper<Consumer> {  *//添加自定义方法，用来完成批量删除   /\*\*  \* 批量删除用户  \* @param ids  \* @return  \*/* int deleteByIds(String ids);  }  ConsumerMapper.xml:  *<?*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.example.onlinemusicservice.mapper.ConsumerMapper">  <resultMap id="BaseResultMap" type="com.example.onlinemusicservice.model.domain.Consumer">  <result column="id" property="id" jdbcType="INTEGER" />  <result column="username" property="username" jdbcType="VARCHAR" />  <result column="password" property="password" jdbcType="VARCHAR" />  <result column="sex" property="sex" jdbcType="TINYINT" />  <result column="phone\_num" property="phoneNum" jdbcType="CHAR" />  <result column="email" property="email" jdbcType="CHAR" />  <result column="birth" property="birth" jdbcType="TIMESTAMP" />  <result column="introduction" property="introduction" jdbcType="VARCHAR"  />  <result column="location" property="location" jdbcType="VARCHAR" />  <result column="avator" property="avator" jdbcType="VARCHAR" />  <result column="create\_time" property="createTime" jdbcType="TIMESTAMP"  />  <result column="update\_time" property="updateTime" jdbcType="TIMESTAMP"  />  </resultMap> *<!--根据id批量删除用户-->* <delete id="deleteByIds">  delete from consumer where id in (${ids})  </delete>   <sql id="Base\_Column\_List">  id, username, password, sex, phone\_num, email, birth, introduction, location, avator, create\_time, update\_time  </sql> </mapper>  ConsumerService.java:  package com.example.onlinemusicservice.service;  import com.baomidou.mybatisplus.core.metadata.IPage; import com.baomidou.mybatisplus.extension.service.IService; import com.example.onlinemusicservice.common.R; import com.example.onlinemusicservice.model.domain.Consumer;  */\*\*  \* admin-用户管理（service层接口）  \*/* public interface ConsumerService extends IService<Consumer> {  */\*\*  \* 管理界面调用：返回所有用户进行分页管理接口  \* @param page  \* @return  \*/* R allUser(IPage page);    */\*\*  \* 通过id删除单个用户接口  \* @param id  \* @return  \*/* R deleteUser(Integer id);   */\*\*  \* 批量删除用户接口  \* @param ids  \* @return  \*/* R deleteUsers(String[] ids);  }  ConsumerServiceImpl.java:  package com.example.onlinemusicservice.service.Impl;  import com.baomidou.mybatisplus.core.metadata.IPage; import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl; import com.example.onlinemusicservice.common.R; import com.example.onlinemusicservice.mapper.ConsumerMapper; import com.example.onlinemusicservice.model.domain.Consumer; import com.example.onlinemusicservice.service.ConsumerService; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Service;  */\*\*  \* admin -用户管理（service层接口实现类）  \*/* @Service public class ConsumerServiceImpl extends ServiceImpl<ConsumerMapper, Consumer> implements ConsumerService {  @Autowired  private ConsumerMapper consumerMapper;   */\*\*  \* 查询所有用户信息进行分页实现方法  \* @param page  \* @return  \*/* @Override  public R allUser(IPage page) {  return R.*success*("查询成功",consumerMapper.selectPage(page,null));  }   */\*\*  \* 删除用户  \*/* @Override  public R deleteUser(Integer id) {  if (consumerMapper.deleteById(id) > 0) {  return R.*success*("删除成功");  } else {  return R.*error*("删除失败");  }  }   */\*\*  \* 批量删除多个用户  \*/* @Override  public R deleteUsers(String[] ids) {  *//把数组转成字符串  //把数组中参数组织成SQL语句需要的格式* StringBuffer sb = new StringBuffer();  for (String i : ids) {  sb.append(i + ",");  }  *//截掉最后一个逗号* sb.deleteCharAt(sb.length() - 1);  if (consumerMapper.deleteByIds(sb.toString()) > 0) {  return R.*success*("删除成功");  } else {  return R.*error*("删除失败");  }  }   } | | | | | | |
| Running Result运行结果: | | | | | | |
| Exercise练习: Exercise2 | | | | | | |
| Request要求:  (Problem Statement)关于问题的描述  Practice2 分页查询 | | | | | | |
| Solution解决:   1. type 创建分页配置类 MyBatisPlusPaginationConfig.java   @Configuration public class MyBatisPlusPaginationConfig {  @Bean  public MybatisPlusInterceptor mybatisPlusInterceptor() {  *//分页插件* PaginationInnerInterceptor pageInterceptor = new PaginationInnerInterceptor();  *//数据库类型* pageInterceptor.setDbType(DbType.*MYSQL*);  *//总拦截器* MybatisPlusInterceptor interceptors = new MybatisPlusInterceptor();  *//添加到插件列表* interceptors.addInnerInterceptor(pageInterceptor);  return interceptors;  } }  2.type修改ConsumerController.java  @GetMapping("/user") public R allUser(@RequestParam(defaultValue = "1") int curPage, @RequestParam(defaultValue = "1") int pageSize) {  *//分* IPage<Consumer> page = new Page<>();  *//设置当前页* page.setCurrent(curPage);  *//设置每页显示数据条数* page.setSize(pageSize);  return consumerService.allUser(page); }   1. type 修改服务接口ConsumerService.java   R allUser(IPage page);   1. type 修改服务实现类ConsumerServiceImpl.java   */\*\*  \* 查询所有用户信息进行分页实现方法  \* @param page  \* @return  \*/* @Override public R allUser(IPage page) {  return R.*success*("查询成功",consumerMapper.selectPage(page,null)); }  4.type 启动前端和后端 | | | | | | |
| Source code源代码:  ConsumerController.java  package com.example.onlinemusicservice.controller;  import com.baomidou.mybatisplus.core.metadata.IPage; import com.baomidou.mybatisplus.extension.plugins.pagination.Page; import com.example.onlinemusicservice.common.R; import com.example.onlinemusicservice.model.domain.Consumer; import com.example.onlinemusicservice.service.ConsumerService; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.web.bind.annotation.DeleteMapping; import org.springframework.web.bind.annotation.GetMapping; import org.springframework.web.bind.annotation.RequestParam; import org.springframework.web.bind.annotation.RestController;  */\*\*  \* admin-用户管理控制类（controller层）  \*/* @RestController public class ConsumerController {  @Autowired  private ConsumerService consumerService;   */\*\*  \* 管理界面调用：返回所有用户进行分页管理  \* @param curPage 当前页  \* @param pageSize 页的大小  \* @return  \*/* @GetMapping("/user")  public R allUser(@RequestParam(defaultValue = "1") int curPage, @RequestParam(defaultValue = "1") int pageSize) {  *//分* IPage<Consumer> page = new Page<>();  *//设置当前页* page.setCurrent(curPage);  *//设置每页显示数据条数* page.setSize(pageSize);  return consumerService.allUser(page);  }   */\*\*  \* 删除用户（根据id删除）  \*/* @GetMapping("/user/delete")  public R deleteUser(@RequestParam int id) {  return consumerService.deleteUser(id);  }   */\*\*  \* 批量删除多个用户  \*  \* @param id  \* @return  \*/* @DeleteMapping("user/deleteIds")  public R deleteUsers(@RequestParam String[] id) {  return consumerService.deleteUsers(id);  } }  ConsumerMapper:  package com.example.onlinemusicservice.mapper;  import com.baomidou.mybatisplus.core.mapper.BaseMapper; import com.example.onlinemusicservice.model.domain.Consumer;  */\*\*  \* admin-用户管理（Mapper层接口）  \*/* public interface ConsumerMapper extends BaseMapper<Consumer> {  *//添加自定义方法，用来完成批量删除   /\*\*  \* 批量删除用户  \* @param ids  \* @return  \*/* int deleteByIds(String ids);  }  ConsumerMapper.xml:  *<?*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.example.onlinemusicservice.mapper.ConsumerMapper">  <resultMap id="BaseResultMap" type="com.example.onlinemusicservice.model.domain.Consumer">  <result column="id" property="id" jdbcType="INTEGER" />  <result column="username" property="username" jdbcType="VARCHAR" />  <result column="password" property="password" jdbcType="VARCHAR" />  <result column="sex" property="sex" jdbcType="TINYINT" />  <result column="phone\_num" property="phoneNum" jdbcType="CHAR" />  <result column="email" property="email" jdbcType="CHAR" />  <result column="birth" property="birth" jdbcType="TIMESTAMP" />  <result column="introduction" property="introduction" jdbcType="VARCHAR"  />  <result column="location" property="location" jdbcType="VARCHAR" />  <result column="avator" property="avator" jdbcType="VARCHAR" />  <result column="create\_time" property="createTime" jdbcType="TIMESTAMP"  />  <result column="update\_time" property="updateTime" jdbcType="TIMESTAMP"  />  </resultMap> *<!--根据id批量删除用户-->* <delete id="deleteByIds">  delete from consumer where id in (${ids})  </delete>   <sql id="Base\_Column\_List">  id, username, password, sex, phone\_num, email, birth, introduction, location, avator, create\_time, update\_time  </sql> </mapper>  ConsumerService.java:  package com.example.onlinemusicservice.service;  import com.baomidou.mybatisplus.core.metadata.IPage; import com.baomidou.mybatisplus.extension.service.IService; import com.example.onlinemusicservice.common.R; import com.example.onlinemusicservice.model.domain.Consumer;  */\*\*  \* admin-用户管理（service层接口）  \*/* public interface ConsumerService extends IService<Consumer> {  */\*\*  \* 管理界面调用：返回所有用户进行分页管理接口  \* @param page  \* @return  \*/* R allUser(IPage page);    */\*\*  \* 通过id删除单个用户接口  \* @param id  \* @return  \*/* R deleteUser(Integer id);   */\*\*  \* 批量删除用户接口  \* @param ids  \* @return  \*/* R deleteUsers(String[] ids);  }  ConsumerServiceImpl.java:  package com.example.onlinemusicservice.service.Impl;  import com.baomidou.mybatisplus.core.metadata.IPage; import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl; import com.example.onlinemusicservice.common.R; import com.example.onlinemusicservice.mapper.ConsumerMapper; import com.example.onlinemusicservice.model.domain.Consumer; import com.example.onlinemusicservice.service.ConsumerService; import org.springframework.beans.factory.annotation.Autowired; import org.springframework.stereotype.Service;  */\*\*  \* admin -用户管理（service层接口实现类）  \*/* @Service public class ConsumerServiceImpl extends ServiceImpl<ConsumerMapper, Consumer> implements ConsumerService {  @Autowired  private ConsumerMapper consumerMapper;   */\*\*  \* 查询所有用户信息进行分页实现方法  \* @param page  \* @return  \*/* @Override  public R allUser(IPage page) {  return R.*success*("查询成功",consumerMapper.selectPage(page,null));  }   */\*\*  \* 删除用户  \*/* @Override  public R deleteUser(Integer id) {  if (consumerMapper.deleteById(id) > 0) {  return R.*success*("删除成功");  } else {  return R.*error*("删除失败");  }  }   */\*\*  \* 批量删除多个用户  \*/* @Override  public R deleteUsers(String[] ids) {  *//把数组转成字符串  //把数组中参数组织成SQL语句需要的格式* StringBuffer sb = new StringBuffer();  for (String i : ids) {  sb.append(i + ",");  }  *//截掉最后一个逗号* sb.deleteCharAt(sb.length() - 1);  if (consumerMapper.deleteByIds(sb.toString()) > 0) {  return R.*success*("删除成功");  } else {  return R.*error*("删除失败");  }  }   }   MyBatisPlusPaginationConfig.java  package com.example.onlinemusicservice.config;  import com.baomidou.mybatisplus.annotation.DbType; import com.baomidou.mybatisplus.extension.plugins.MybatisPlusInterceptor; import com.baomidou.mybatisplus.extension.plugins.inner.PaginationInnerInterceptor; import org.springframework.context.annotation.Bean; import org.springframework.context.annotation.Configuration;  */\*\*  \* MyBatis Plus分页配置类  \*/* @Configuration public class MyBatisPlusPaginationConfig {  @Bean  public MybatisPlusInterceptor mybatisPlusInterceptor() {  *//分页插件* PaginationInnerInterceptor pageInterceptor = new PaginationInnerInterceptor();  *//数据库类型* pageInterceptor.setDbType(DbType.*MYSQL*);  *//总拦截器* MybatisPlusInterceptor interceptors = new MybatisPlusInterceptor();  *//添加到插件列表* interceptors.addInnerInterceptor(pageInterceptor);  return interceptors;  } } | | | | | | |
| Running Result运行结果: | | | | | | |
| Experimental  Experience  实验心得 | Problems encountered遇到的问题:  无 | | | | | | |
| Knowledge mastered掌握的知识:学会批量删除 | | | | | | |
| Comments  评语 |  | | | | | | |
| Score 成绩 |  | | | | | | |