<!DOCTYPE html><html lang="zh-CN"><head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>卓越健身管理系统</title> <link rel="stylesheet"> <link rel="stylesheet" ></head><body> <header class="rz-header"> <h1>卓越健身管理系统</h1> </header> <div class="rz-container"> <nav class="rz-sidebar"> <ul class="rz-nav-list"> <a ><li class="rz-nav-item active"><span>个人健身档案</span></li></a> <a ><li class="rz-nav-item"><span>课程预约</span></li></a> <a ><li class="rz-nav-item"><span>在线支付</span></li></a> <a ><li class="rz-nav-item"><span>健身计划制定</span></li></a> <a ><li class="rz-nav-item"><span>数据统计与分析</span></li></a> <a ><li class="rz-nav-item"><span>教练管理中心</span></li></a> </ul> </nav> <main class="rz-main-content"> <div id="content" class="rz-card"> <h2>个人健身档案</h2> <form> <div class="form-group"> <label for="height">身高 (cm)</label> <input type="number" id="height" placeholder="请输入您的身高"> </div> <div class="form-group"> <label for="weight">体重 (kg)</label> <input type="number" id="weight" placeholder="请输入您的体重"> </div> <div class="form-group"> <label for="goal">健身目标</label> <textarea id="goal" rows="4" placeholder="请输入您的健身目标"></textarea> </div> <div class="form-group"> <label for="photos">上传健身照片</label> <input type="file" id="photos" multiple> </div> <button type="submit" class="btn">保存档案</button> </form> </div> </main> </div></body></html>

<!DOCTYPE html><html lang="zh-CN"><head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>卓越健身管理系统</title> <link rel="stylesheet"></head><body> <header class="rz-header"> <h1>卓越健身管理系统</h1> </header> <div class="rz-container"> <nav class="rz-sidebar"> <ul class="rz-nav-list"> <a ><li class="rz-nav-item"><span>个人健身档案</span></li></a> <a ><li class="rz-nav-item active"><span>课程预约</span></li></a> <a ><li class="rz-nav-item"><span>在线支付</span></li></a> <a ><li class="rz-nav-item"><span>健身计划制定</span></li></a> <a ><li class="rz-nav-item"><span>数据统计与分析</span></li></a> <a ><li class="rz-nav-item"><span>教练管理中心</span></li></a> </ul> </nav> <main class="rz-main-content"> <div id="content" class="rz-card"> <h2>可用健身课程</h2> <div class="course-item"> <h3>瑜伽课程</h3> <p>时间：每周一 9:00 - 10:00</p> <p>地点：健身房A区</p> <p>教练：李老师</p> <p>课程内容：基础瑜伽，适合初学者</p> <p>剩余名额：5</p> <button class="btn-appointment">预约课程</button> </div> <div class="course-item"> <h3>力量训练</h3> <p>时间：每周三 18:00 - 19:00</p> <p>地点：健身房B区</p> <p>教练：王教练</p> <p>课程内容：全身力量训练</p> <p>剩余名额：3</p> <button class="btn-appointment">预约课程</button> </div> <div class="course-item"> <h3>有氧运动</h3> <p>时间：每周五 17:00 - 18:00</p> <p>地点：健身房C区</p> <p>教练：张教练</p> <p>课程内容：高强度有氧训练</p> <p>剩余名额：2</p> <button class="btn-appointment">预约课程</button> </div> <!-- 可以继续添加更多课程 --> </div> </main> </div></body></html>

<!DOCTYPE html><html lang="zh-CN"><head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>卓越健身管理系统</title> <link rel="stylesheet"> <link rel="stylesheet" ></head><body> <header class="rz-header"> <h1>卓越健身管理系统</h1> </header> <div class="rz-container"> <nav class="rz-sidebar"> <ul class="rz-nav-list"> <a ><li class="rz-nav-item"><span>个人健身档案</span></li></a> <a ><li class="rz-nav-item"><span>课程预约</span></li></a> <a ><li class="rz-nav-item active"><span>在线支付</span></li></a> <a ><li class="rz-nav-item"><span>健身计划制定</span></li></a> <a ><li class="rz-nav-item"><span>数据统计与分析</span></li></a> <a ><li class="rz-nav-item"><span>教练管理中心</span></li></a> </ul> </nav> <main class="rz-main-content"> <div id="content" class="rz-card"> <h2>在线支付</h2> <p>本页面用于处理用户的在线支付请求。用户在选择课程或购买健身计划后，可通过该页面进行安全的支付操作。系统支持多种支付方式，用户需输入相关支付信息并确认支付。成功支付后，用户将收到支付凭证，并可在个人档案中查看支付记录。</p> <form class="payment-form"> <label for="name">姓名:</label> <input type="text" id="name" name="name" required> <label for="amount">支付金额:</label> <input type="number" id="amount" name="amount" required> <label for="payment-method">支付方式:</label> <select id="payment-method" name="payment-method" required> <option value="credit-card">信用卡</option> <option value="paypal">Paypal</option> <option value="wechat">微信支付</option> <option value="alipay">支付宝</option> </select> <label for="card-number">卡号:</label> <input type="text" id="card-number" name="card-number" required> <button type="submit">确认支付</button> </form> </div> </main> </div></body></html>

<!DOCTYPE html><html lang="zh-CN"><head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>卓越健身管理系统</title> <link rel="stylesheet"> <link rel="stylesheet" ></head><body> <header class="rz-header"> <h1>卓越健身管理系统</h1> </header> <div class="rz-container"> <nav class="rz-sidebar"> <ul class="rz-nav-list"> <a ><li class="rz-nav-item"><span>个人健身档案</span></li></a> <a ><li class="rz-nav-item"><span>课程预约</span></li></a> <a ><li class="rz-nav-item"><span>在线支付</span></li></a> <a ><li class="rz-nav-item active"><span>健身计划制定</span></li></a> <a ><li class="rz-nav-item"><span>数据统计与分析</span></li></a> <a ><li class="rz-nav-item"><span>教练管理中心</span></li></a> </ul> </nav> <main class="rz-main-content"> <div id="content" class="rz-card"> <h2>健身计划制定</h2> <p>在此页面，用户可以根据自己的健身目标和身体状况接受个性化的健身计划制定。系统会分析用户的健身档案并提供适合的训练方案，包括每天的锻炼内容和时间安排。用户可根据计划进行锻炼，并记录训练进度，以便随时调整方案。</p> <form> <div class="form-group"> <label for="goal">健身目标</label> <select id="goal" name="goal"> <option value="减脂">减脂</option> <option value="增肌">增肌</option> <option value="保持">保持</option> </select> </div> <div class="form-group"> <label for="level">健身水平</label> <select id="level" name="level"> <option value="初学者">初学者</option> <option value="中级">中级</option> <option value="高级">高级</option> </select> </div> <div class="form-group"> <label for="duration">锻炼时长（分钟）</label> <input type="number" id="duration" name="duration" required> </div> <div class="form-group"> <label for="notes">备注</label> <textarea id="notes" name="notes" rows="4"></textarea> </div> <button type="submit" class="button">制定计划</button> </form> <div class="train-log"> <h3>训练记录</h3> <p>记录您的训练进度:</p> <ul> <li>日期: 2023-10-01 | 锻炼内容: 跑步 30分钟</li> <li>日期: 2023-10-02 | 锻炼内容: 力量训练 45分钟</li> </ul> </div> </div> </main> </div></body></html>

<!DOCTYPE html><html lang="zh-CN"><head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>卓越健身管理系统</title> <link rel="stylesheet"> <link rel="stylesheet" ></head><body> <header class="rz-header"> <h1>卓越健身管理系统</h1> </header> <div class="rz-container"> <nav class="rz-sidebar"> <ul class="rz-nav-list"> <a ><li class="rz-nav-item"><span>个人健身档案</span></li></a> <a ><li class="rz-nav-item"><span>课程预约</span></li></a> <a ><li class="rz-nav-item"><span>在线支付</span></li></a> <a ><li class="rz-nav-item"><span>健身计划制定</span></li></a> <a ><li class="rz-nav-item active"><span>数据统计与分析</span></li></a> <a ><li class="rz-nav-item"><span>教练管理中心</span></li></a> </ul> </nav> <main class="rz-main-content"> <div id="content" class="rz-card"> <h2>健身进度统计与分析</h2> <p>在此页面，您可以查看自己的锻炼记录、体重变化、卡路里消耗等数据，通过图表形式直观了解健身效果。</p> <div class="chart"> <!-- 图表区域 --> <p style="text-align: center; padding-top: 130px;">图表将在此呈现</p> </div> <h3>改进建议</h3> <p>根据您的数据，系统将提供个性化的改进建议，帮助您优化健身策略，以实现更好的健康效果。</p> </div> </main> </div></body></html>

<!DOCTYPE html><html lang="zh-CN"><head> <meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>卓越健身管理系统</title> <link rel="stylesheet"> <link rel="stylesheet" ></head><body> <header class="rz-header"> <h1>卓越健身管理系统</h1> </header> <div class="rz-container"> <nav class="rz-sidebar"> <ul class="rz-nav-list"> <a ><li class="rz-nav-item"><span>个人健身档案</span></li></a> <a ><li class="rz-nav-item"><span>课程预约</span></li></a> <a ><li class="rz-nav-item"><span>在线支付</span></li></a> <a ><li class="rz-nav-item"><span>健身计划制定</span></li></a> <a ><li class="rz-nav-item"><span>数据统计与分析</span></li></a> <a ><li class="rz-nav-item active"><span>教练管理中心</span></li></a> </ul> </nav> <main class="rz-main-content"> <div id="content" class="rz-card"> <h2>教练管理中心</h2> <h3>学员信息管理</h3> <table class="rz-table"> <thead> <tr> <th>学员姓名</th> <th>健身档案</th> <th>进度记录</th> <th>操作</th> </tr> </thead> <tbody> <tr> <td>张三</td> <td><a >查看档案</a></td> <td><a >查看记录</a></td> <td> <button>编辑</button> <button>删除</button> </td> </tr> <tr> <td>李四</td> <td><a >查看档案</a></td> <td><a >查看记录</a></td> <td> <button>编辑</button> <button>删除</button> </td> </tr> </tbody> </table> <h3>课程安排</h3> <button>发布新课程</button> <button>调整课程时间</button> </div> </main> </div></body></html>

/\*\* \* @description users \* @author 大大的盆 \*/@RestController@RequestMapping(value = "/users")public class UsersController { @Resource private UsersService usersService; /\*\* \* 新增 \* @author 大大的盆 \*\*/ @RequestMapping("/insert") public Object insert(Users users){ return usersService.insert(users); } /\*\* \* 刪除 \* @author 大大的盆 \*\*/ @RequestMapping("/delete") public Object delete(int id){ return usersService.delete(id); } /\*\* \* 更新 \* @author 大大的盆 \*\*/ @RequestMapping("/update") public Object update(Users users){ return usersService.update(users); } /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ @RequestMapping("/load") public Object load(int id){ return usersService.load(id); } /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ @RequestMapping("/pageList") public Map<String, Object> pageList(@RequestParam(required = false, defaultValue = "0") int offset, @RequestParam(required = false, defaultValue = "10") int pagesize) { return usersService.pageList(offset, pagesize); }}/\*\* \* @description users \* @author 大大的盆 \*/public interface UsersService { /\*\* \* 新增 \*/ public Object insert(Users users); /\*\* \* 删除 \*/ public Object delete(int id); /\*\* \* 更新 \*/ public Object update(Users users); /\*\* \* 根据主键 id 查询 \*/ public Users load(int id); /\*\* \* 分页查询 \*/ public Map<String,Object> pageList(int offset, int pagesize);}/\*\* \* @description users \* @author 大大的盆 \*/@Servicepublic class UsersServiceImpl implements UsersService { @Resource private UsersMapper usersMapper; @Override public Object insert(Users users) { // valid if (users == null) { return ReturnT.error("必要参数缺失"); } usersMapper.insert(users); return ReturnT.success(); } @Override public Object delete(int id) { int ret = usersMapper.delete(id); return ret>0?ReturnT.success():ReturnT.error(); } @Override public Object update(Users users) { int ret = usersMapper.update(users); return ret>0?ReturnT.success():ReturnT.error(); } @Override public Users load(int id) { return usersMapper.load(id); } @Override public Map<String,Object> pageList(int offset, int pagesize) { List<Users> pageList = usersMapper.pageList(offset, pagesize); int totalCount = usersMapper.pageListCount(offset, pagesize); // result Map<String, Object> result = new HashMap<String, Object>(); result.put("pageList", pageList); result.put("totalCount", totalCount); return result; }}/\*\* \* @description users \* @author 大大的盆 \*/@Mapper@Repositorypublic interface UsersMapper { /\*\* \* 新增 \* @author 大大的盆 \*\*/ int insert(Users users); /\*\* \* 刪除 \* @author 大大的盆 \*\*/ int delete(int id); /\*\* \* 更新 \* @author 大大的盆 \*\*/ int update(Users users); /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ Users load(int id); /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ List<Users> pageList(int offset,int pagesize); /\*\* \* 查询 分页查询 count \* @author 大大的盆 \*\*/ int pageListCount(int offset,int pagesize);}<?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="HBJZGCXY.com.dao.UsersMapper"> <resultMap id="BaseResultMap" type="HBJZGCXY.com.entity.Users" > <result column="username" property="username" /> <result column="password" property="password" /> <result column="email" property="email" /> <result column="create\_time" property="createTime" /> <result column="create\_user" property="createUser" /> <result column="update\_time" property="updateTime" /> <result column="update\_user" property="updateUser" /> </resultMap> <sql id="Base\_Column\_List"> username, password, email, create\_time, create\_user, update\_time, update\_user </sql> <insert id="insert" useGeneratedKeys="true" keyColumn="id" keyProperty="id" parameterType="HBJZGCXY.com.entity.Users"> INSERT INTO users <trim prefix="(" suffix=")" suffixOverrides=","> <if test="null != username and '' != username"> username, </if> <if test="null != password and '' != password"> password, </if> <if test="null != email and '' != email"> email, </if> <if test="null != createTime and '' != createTime"> create\_time, </if> <if test="null != createUser and '' != createUser"> create\_user, </if> <if test="null != updateTime and '' != updateTime"> update\_time, </if> <if test="null != updateUser and '' != updateUser"> update\_user </if> </trim> <trim prefix="values (" suffix=")" suffixOverrides=","> <if test="null != username and '' != username"> #{username}, </if> <if test="null != password and '' != password"> #{password}, </if> <if test="null != email and '' != email"> #{email}, </if> <if test="null != createTime and '' != createTime"> #{createTime}, </if> <if test="null != createUser and '' != createUser"> #{createUser}, </if> <if test="null != updateTime and '' != updateTime"> #{updateTime}, </if> <if test="null != updateUser and '' != updateUser"> #{updateUser} </if> </trim> </insert> <delete id="delete" > DELETE FROM users WHERE id = #{id} </delete> <update id="update" parameterType="HBJZGCXY.com.entity.Users"> UPDATE users <set> <if test="null != username and '' != username">username = #{username},</if> <if test="null != password and '' != password">password = #{password},</if> <if test="null != email and '' != email">email = #{email},</if> <if test="null != createTime and '' != createTime">create\_time = #{createTime},</if> <if test="null != createUser and '' != createUser">create\_user = #{createUser},</if> <if test="null != updateTime and '' != updateTime">update\_time = #{updateTime},</if> <if test="null != updateUser and '' != updateUser">update\_user = #{updateUser}</if> </set> WHERE id = #{id} </update> <select id="load" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM users WHERE id = #{id} </select> <select id="pageList" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM users LIMIT #{offset}, #{pageSize} </select> <select id="pageListCount" resultType="java.lang.Integer"> SELECT count(1) FROM users </select></mapper>/\*\* \* @description users \* @author 大大的盆 \*/public class Users implements Serializable { private static final long serialVersionUID = 1L; private String username; private String password; private String email; private String createTime; private String createUser; private String updateTime; private String updateUser; public Users() { } public String getUsername() { return username; } public void setUsername(String username) { this.username = username; } public String getPassword() { return password; } public void setPassword(String password) { this.password = password; } public String getEmail() { return email; } public void setEmail(String email) { this.email = email; } public String getCreateTime() { return createTime; } public void setCreateTime(String createTime) { this.createTime = createTime; } public String getCreateUser() { return createUser; } public void setCreateUser(String createUser) { this.createUser = createUser; } public String getUpdateTime() { return updateTime; } public void setUpdateTime(String updateTime) { this.updateTime = updateTime; } public String getUpdateUser() { return updateUser; } public void setUpdateUser(String updateUser) { this.updateUser = updateUser; }}<template> <div class="mod-config"> <el-form :inline="true" :model="dataForm" @keyup.enter.native="getDataList()"> <el-form-item> <el-input v-model="dataForm.key" placeholder="参数名" clearable></el-input> </el-form-item> <el-form-item> <el-button @click="getDataList()">查询</el-button> <el-button v-if="isAuth('generator:users:save')" type="primary" @click="addOrUpdateHandle()">新增</el-button> <el-button v-if="isAuth('generator:users:delete')" type="danger" @click="deleteHandle()" :disabled="dataListSelections.length <= 0">批量删除</el-button> </el-form-item> </el-form> <el-table :data="dataList" border v-loading="dataListLoading" @selection-change="selectionChangeHandle" style="width: 100%;"> <el-table-column type="selection" header-align="center" align="center" width="50"> </el-table-column> <el-table-column prop="username" header-align="center" align="center" label="username"> </el-table-column> <el-table-column prop="password" header-align="center" align="center" label="password"> </el-table-column> <el-table-column prop="email" header-align="center" align="center" label="email"> </el-table-column> <el-table-column prop="createTime" header-align="center" align="center" label="create\_time"> </el-table-column> <el-table-column prop="createUser" header-align="center" align="center" label="create\_user"> </el-table-column> <el-table-column prop="updateTime" header-align="center" align="center" label="update\_time"> </el-table-column> <el-table-column prop="updateUser" header-align="center" align="center" label="update\_user"> </el-table-column> <el-table-column fixed="right" header-align="center" align="center" width="150" label="操作"> <template slot-scope="scope"> <el-button type="text" size="small" @click="addOrUpdateHandle(scope.row.usersId)">修改</el-button> <el-button type="text" size="small" @click="deleteHandle(scope.row.usersId)">删除</el-button> </template> </el-table-column> </el-table> <el-pagination @size-change="sizeChangeHandle" @current-change="currentChangeHandle" :current-page="pageIndex" :page-sizes="[10, 20, 50, 100]" :page-size="pageSize" :total="totalPage" layout="total, sizes, prev, pager, next, jumper"> </el-pagination> <!-- 弹窗, 新增 / 修改 --> <add-or-update v-if="addOrUpdateVisible" ref="addOrUpdate" @refreshDataList="getDataList"></add-or-update> </div></template><script> import AddOrUpdate from './users-add-or-update' export default { data () { return { dataForm: { key: '' }, dataList: [], pageIndex: 1, pageSize: 10, totalPage: 0, dataListLoading: false, dataListSelections: [], addOrUpdateVisible: false } }, components: { AddOrUpdate }, activated () { this.getDataList() }, methods: { // 获取数据列表 getDataList () { this.dataListLoading = true this.$http({ url: this.$http.adornUrl('/generator/users/list'), method: 'get', params: this.$http.adornParams({ 'page': this.pageIndex, 'limit': this.pageSize, 'key': this.dataForm.key }) }).then(({data}) => { if (data && data.code === 0) { this.dataList = data.page.list this.totalPage = data.page.totalCount } else { this.dataList = [] this.totalPage = 0 } this.dataListLoading = false }) }, // 每页数 sizeChangeHandle (val) { this.pageSize = val this.pageIndex = 1 this.getDataList() }, // 当前页 currentChangeHandle (val) { this.pageIndex = val this.getDataList() }, // 多选 selectionChangeHandle (val) { this.dataListSelections = val }, // 新增 / 修改 addOrUpdateHandle (id) { this.addOrUpdateVisible = true this.$nextTick(() => { this.$refs.addOrUpdate.init(id) }) }, // 删除 deleteHandle (id) { var ids = id ? [id] : this.dataListSelections.map(item => { return item.usersId }) this.$confirm(`确定对[id=${ids.join(',')}]进行[${id ? '删除' : '批量删除'}]操作?`, '提示', { confirmButtonText: '确定', cancelButtonText: '取消', type: 'warning' }).then(() => { this.$http({ url: this.$http.adornUrl('/generator/users/delete'), method: 'post', data: this.$http.adornData(ids, false) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.getDataList() } }) } else { this.$message.error(data.msg) } }) }) } } }</script><template> <el-dialog :title="!dataForm.usersId ? '新增' : '修改'" :close-on-click-modal="false" :visible.sync="visible"> <el-form :model="dataForm" :rules="dataRule" ref="dataForm" @keyup.enter.native="dataFormSubmit()" label-width="80px"> <el-form-item label="username" prop="username"> <el-input v-model="dataForm.username" placeholder="username"></el-input> </el-form-item> <el-form-item label="password" prop="password"> <el-input v-model="dataForm.password" placeholder="password"></el-input> </el-form-item> <el-form-item label="email" prop="email"> <el-input v-model="dataForm.email" placeholder="email"></el-input> </el-form-item> <el-form-item label="create\_time" prop="createTime"> <el-input v-model="dataForm.createTime" placeholder="create\_time"></el-input> </el-form-item> <el-form-item label="create\_user" prop="createUser"> <el-input v-model="dataForm.createUser" placeholder="create\_user"></el-input> </el-form-item> <el-form-item label="update\_time" prop="updateTime"> <el-input v-model="dataForm.updateTime" placeholder="update\_time"></el-input> </el-form-item> <el-form-item label="update\_user" prop="updateUser"> <el-input v-model="dataForm.updateUser" placeholder="update\_user"></el-input> </el-form-item> </el-form> <span slot="footer" class="dialog-footer"> <el-button @click="visible = false">取消</el-button> <el-button type="primary" @click="dataFormSubmit()">确定</el-button> </span> </el-dialog></template><script> export default { data () { return { visible: false, dataForm: { username: '', password: '', email: '', createTime: '', createUser: '', updateTime: '', updateUser: '' }, dataRule: { username: [{ required: true, message: 'username不能为空', trigger: 'blur' }], password: [{ required: true, message: 'password不能为空', trigger: 'blur' }], email: [{ required: true, message: 'email不能为空', trigger: 'blur' }], createTime: [{ required: true, message: 'create\_time不能为空', trigger: 'blur' }], createUser: [{ required: true, message: 'create\_user不能为空', trigger: 'blur' }], updateTime: [{ required: true, message: 'update\_time不能为空', trigger: 'blur' }], updateUser: [{ required: true, message: 'update\_user不能为空', trigger: 'blur' }] } } }, methods: { init (id) { this.dataForm.usersId = id || 0 this.visible = true this.$nextTick(() => { this.$refs['dataForm'].resetFields() if (this.dataForm.usersId) { this.$http({ url: this.$http.adornUrl(`/generator/users/info/${this.dataForm.usersId}`), method: 'get', params: this.$http.adornParams() }).then(({data}) => { if (data && data.code === 0) { this.dataForm.username = data.users.username this.dataForm.password = data.users.password this.dataForm.email = data.users.email this.dataForm.createTime = data.users.createTime this.dataForm.createUser = data.users.createUser this.dataForm.updateTime = data.users.updateTime this.dataForm.updateUser = data.users.updateUser } }) } }) }, // 表单提交 dataFormSubmit () { this.$refs['dataForm'].validate((valid) => { if (valid) { this.$http({ url: this.$http.adornUrl(`/generator/users/${this.dataForm.usersId? 'save' : 'update'}`), method: 'post', data: this.$http.adornData({ 'username': 'username' || undefined, 'password': 'password' || undefined, 'email': 'email' || undefined, 'createTime': 'createTime' || undefined, 'createUser': 'createUser' || undefined, 'updateTime': 'updateTime' || undefined, 'updateUser': 'updateUser' || undefined }) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.visible = false this.$emit('refreshDataList') } }) } else { this.$message.error(data.msg) } }) } }) } } }</script>/\*\* \* @description fitness\_courses \* @author 大大的盆 \*/@RestController@RequestMapping(value = "/fitnessCourses")public class FitnessCoursesController { @Resource private FitnessCoursesService fitnessCoursesService; /\*\* \* 新增 \* @author 大大的盆 \*\*/ @RequestMapping("/insert") public Object insert(FitnessCourses fitnessCourses){ return fitnessCoursesService.insert(fitnessCourses); } /\*\* \* 刪除 \* @author 大大的盆 \*\*/ @RequestMapping("/delete") public Object delete(int id){ return fitnessCoursesService.delete(id); } /\*\* \* 更新 \* @author 大大的盆 \*\*/ @RequestMapping("/update") public Object update(FitnessCourses fitnessCourses){ return fitnessCoursesService.update(fitnessCourses); } /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ @RequestMapping("/load") public Object load(int id){ return fitnessCoursesService.load(id); } /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ @RequestMapping("/pageList") public Map<String, Object> pageList(@RequestParam(required = false, defaultValue = "0") int offset, @RequestParam(required = false, defaultValue = "10") int pagesize) { return fitnessCoursesService.pageList(offset, pagesize); }}/\*\* \* @description fitness\_courses \* @author 大大的盆 \*/public interface FitnessCoursesService { /\*\* \* 新增 \*/ public Object insert(FitnessCourses fitnessCourses); /\*\* \* 删除 \*/ public Object delete(int id); /\*\* \* 更新 \*/ public Object update(FitnessCourses fitnessCourses); /\*\* \* 根据主键 id 查询 \*/ public FitnessCourses load(int id); /\*\* \* 分页查询 \*/ public Map<String,Object> pageList(int offset, int pagesize);}/\*\* \* @description fitness\_courses \* @author 大大的盆 \*/@Servicepublic class FitnessCoursesServiceImpl implements FitnessCoursesService { @Resource private FitnessCoursesMapper fitnessCoursesMapper; @Override public Object insert(FitnessCourses fitnessCourses) { // valid if (fitnessCourses == null) { return ReturnT.error("必要参数缺失"); } fitnessCoursesMapper.insert(fitnessCourses); return ReturnT.success(); } @Override public Object delete(int id) { int ret = fitnessCoursesMapper.delete(id); return ret>0?ReturnT.success():ReturnT.error(); } @Override public Object update(FitnessCourses fitnessCourses) { int ret = fitnessCoursesMapper.update(fitnessCourses); return ret>0?ReturnT.success():ReturnT.error(); } @Override public FitnessCourses load(int id) { return fitnessCoursesMapper.load(id); } @Override public Map<String,Object> pageList(int offset, int pagesize) { List<FitnessCourses> pageList = fitnessCoursesMapper.pageList(offset, pagesize); int totalCount = fitnessCoursesMapper.pageListCount(offset, pagesize); // result Map<String, Object> result = new HashMap<String, Object>(); result.put("pageList", pageList); result.put("totalCount", totalCount); return result; }}/\*\* \* @description fitness\_courses \* @author 大大的盆 \*/@Mapper@Repositorypublic interface FitnessCoursesMapper { /\*\* \* 新增 \* @author 大大的盆 \*\*/ int insert(FitnessCourses fitnessCourses); /\*\* \* 刪除 \* @author 大大的盆 \*\*/ int delete(int id); /\*\* \* 更新 \* @author 大大的盆 \*\*/ int update(FitnessCourses fitnessCourses); /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ FitnessCourses load(int id); /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ List<FitnessCourses> pageList(int offset,int pagesize); /\*\* \* 查询 分页查询 count \* @author 大大的盆 \*\*/ int pageListCount(int offset,int pagesize);}<?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="HBJZGCXY.com.dao.FitnessCoursesMapper"> <resultMap id="BaseResultMap" type="HBJZGCXY.com.entity.FitnessCourses" > <result column="course\_name" property="courseName" /> <result column="description" property="description" /> <result column="duration" property="duration" /> <result column="--" property="" /> <result column="--" property="" /> <result column="create\_user" property="createUser" /> <result column="update\_time" property="updateTime" /> <result column="update\_user" property="updateUser" /> </resultMap> <sql id="Base\_Column\_List"> course\_name, description, duration, --, --, create\_user, update\_time, update\_user </sql> <insert id="insert" useGeneratedKeys="true" keyColumn="id" keyProperty="id" parameterType="HBJZGCXY.com.entity.FitnessCourses"> INSERT INTO fitness\_courses <trim prefix="(" suffix=")" suffixOverrides=","> <if test="null != courseName and '' != courseName"> course\_name, </if> <if test="null != description and '' != description"> description, </if> <if test="null != duration and '' != duration"> duration, </if> <if test="null != and '' != "> --, </if> <if test="null != and '' != "> --, </if> <if test="null != createUser and '' != createUser"> create\_user, </if> <if test="null != updateTime and '' != updateTime"> update\_time, </if> <if test="null != updateUser and '' != updateUser"> update\_user </if> </trim> <trim prefix="values (" suffix=")" suffixOverrides=","> <if test="null != courseName and '' != courseName"> #{courseName}, </if> <if test="null != description and '' != description"> #{description}, </if> <if test="null != duration and '' != duration"> #{duration}, </if> <if test="null != and '' != "> #{}, </if> <if test="null != and '' != "> #{}, </if> <if test="null != createUser and '' != createUser"> #{createUser}, </if> <if test="null != updateTime and '' != updateTime"> #{updateTime}, </if> <if test="null != updateUser and '' != updateUser"> #{updateUser} </if> </trim> </insert> <delete id="delete" > DELETE FROM fitness\_courses WHERE id = #{id} </delete> <update id="update" parameterType="HBJZGCXY.com.entity.FitnessCourses"> UPDATE fitness\_courses <set> <if test="null != courseName and '' != courseName">course\_name = #{courseName},</if> <if test="null != description and '' != description">description = #{description},</if> <if test="null != duration and '' != duration">duration = #{duration},</if> <if test="null != and '' != ">-- = #{},</if> <if test="null != and '' != ">-- = #{},</if> <if test="null != createUser and '' != createUser">create\_user = #{createUser},</if> <if test="null != updateTime and '' != updateTime">update\_time = #{updateTime},</if> <if test="null != updateUser and '' != updateUser">update\_user = #{updateUser}</if> </set> WHERE id = #{id} </update> <select id="load" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM fitness\_courses WHERE id = #{id} </select> <select id="pageList" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM fitness\_courses LIMIT #{offset}, #{pageSize} </select> <select id="pageListCount" resultType="java.lang.Integer"> SELECT count(1) FROM fitness\_courses </select></mapper>/\*\* \* @description fitness\_courses \* @author 大大的盆 \*/public class FitnessCourses implements Serializable { private static final long serialVersionUID = 1L; private String courseName; private String description; private String duration; private String ; private String ; private String createUser; private String updateTime; private String updateUser; public FitnessCourses() { } public String getCourseName() { return courseName; } public void setCourseName(String courseName) { this.courseName = courseName; } public String getDescription() { return description; } public void setDescription(String description) { this.description = description; } public String getDuration() { return duration; } public void setDuration(String duration) { this.duration = duration; } public String get() { return ; } public void set(String ) { this. = ; } public String get() { return ; } public void set(String ) { this. = ; } public String getCreateUser() { return createUser; } public void setCreateUser(String createUser) { this.createUser = createUser; } public String getUpdateTime() { return updateTime; } public void setUpdateTime(String updateTime) { this.updateTime = updateTime; } public String getUpdateUser() { return updateUser; } public void setUpdateUser(String updateUser) { this.updateUser = updateUser; }}<template> <div class="mod-config"> <el-form :inline="true" :model="dataForm" @keyup.enter.native="getDataList()"> <el-form-item> <el-input v-model="dataForm.key" placeholder="参数名" clearable></el-input> </el-form-item> <el-form-item> <el-button @click="getDataList()">查询</el-button> <el-button v-if="isAuth('generator:fitnessCourses:save')" type="primary" @click="addOrUpdateHandle()">新增</el-button> <el-button v-if="isAuth('generator:fitnessCourses:delete')" type="danger" @click="deleteHandle()" :disabled="dataListSelections.length <= 0">批量删除</el-button> </el-form-item> </el-form> <el-table :data="dataList" border v-loading="dataListLoading" @selection-change="selectionChangeHandle" style="width: 100%;"> <el-table-column type="selection" header-align="center" align="center" width="50"> </el-table-column> <el-table-column prop="courseName" header-align="center" align="center" label="course\_name"> </el-table-column> <el-table-column prop="description" header-align="center" align="center" label="description"> </el-table-column> <el-table-column prop="duration" header-align="center" align="center" label="duration"> </el-table-column> <el-table-column prop="" header-align="center" align="center" label="--"> </el-table-column> <el-table-column prop="" header-align="center" align="center" label="--"> </el-table-column> <el-table-column prop="createUser" header-align="center" align="center" label="create\_user"> </el-table-column> <el-table-column prop="updateTime" header-align="center" align="center" label="update\_time"> </el-table-column> <el-table-column prop="updateUser" header-align="center" align="center" label="update\_user"> </el-table-column> <el-table-column fixed="right" header-align="center" align="center" width="150" label="操作"> <template slot-scope="scope"> <el-button type="text" size="small" @click="addOrUpdateHandle(scope.row.fitnessCoursesId)">修改</el-button> <el-button type="text" size="small" @click="deleteHandle(scope.row.fitnessCoursesId)">删除</el-button> </template> </el-table-column> </el-table> <el-pagination @size-change="sizeChangeHandle" @current-change="currentChangeHandle" :current-page="pageIndex" :page-sizes="[10, 20, 50, 100]" :page-size="pageSize" :total="totalPage" layout="total, sizes, prev, pager, next, jumper"> </el-pagination> <!-- 弹窗, 新增 / 修改 --> <add-or-update v-if="addOrUpdateVisible" ref="addOrUpdate" @refreshDataList="getDataList"></add-or-update> </div></template><script> import AddOrUpdate from './fitnessCourses-add-or-update' export default { data () { return { dataForm: { key: '' }, dataList: [], pageIndex: 1, pageSize: 10, totalPage: 0, dataListLoading: false, dataListSelections: [], addOrUpdateVisible: false } }, components: { AddOrUpdate }, activated () { this.getDataList() }, methods: { // 获取数据列表 getDataList () { this.dataListLoading = true this.$http({ url: this.$http.adornUrl('/generator/fitnessCourses/list'), method: 'get', params: this.$http.adornParams({ 'page': this.pageIndex, 'limit': this.pageSize, 'key': this.dataForm.key }) }).then(({data}) => { if (data && data.code === 0) { this.dataList = data.page.list this.totalPage = data.page.totalCount } else { this.dataList = [] this.totalPage = 0 } this.dataListLoading = false }) }, // 每页数 sizeChangeHandle (val) { this.pageSize = val this.pageIndex = 1 this.getDataList() }, // 当前页 currentChangeHandle (val) { this.pageIndex = val this.getDataList() }, // 多选 selectionChangeHandle (val) { this.dataListSelections = val }, // 新增 / 修改 addOrUpdateHandle (id) { this.addOrUpdateVisible = true this.$nextTick(() => { this.$refs.addOrUpdate.init(id) }) }, // 删除 deleteHandle (id) { var ids = id ? [id] : this.dataListSelections.map(item => { return item.fitnessCoursesId }) this.$confirm(`确定对[id=${ids.join(',')}]进行[${id ? '删除' : '批量删除'}]操作?`, '提示', { confirmButtonText: '确定', cancelButtonText: '取消', type: 'warning' }).then(() => { this.$http({ url: this.$http.adornUrl('/generator/fitnessCourses/delete'), method: 'post', data: this.$http.adornData(ids, false) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.getDataList() } }) method: 'get', params: this.$http.adornParams() }).then(({data}) => { if (data && data.code === 0) { this.dataForm.userId = data.reservations.userId this.dataForm.courseId = data.reservations.courseId this.dataForm.reservationTime = data.reservations.reservationTime this.dataForm.status = data.reservations.status this.dataForm.createTime = data.reservations.createTime this.dataForm.createUser = data.reservations.createUser this.dataForm.updateTime = data.reservations.updateTime this.dataForm.updateUser = data.reservations.updateUser } }) } }) }, // 表单提交 dataFormSubmit () { this.$refs['dataForm'].validate((valid) => { if (valid) { this.$http({ url: this.$http.adornUrl(`/generator/reservations/${this.dataForm.reservationsId? 'save' : 'update'}`), method: 'post', data: this.$http.adornData({ 'userId': 'userId' || undefined, 'courseId': 'courseId' || undefined, 'reservationTime': 'reservationTime' || undefined, 'status': 'status' || undefined, 'createTime': 'createTime' || undefined, 'createUser': 'createUser' || undefined, 'updateTime': 'updateTime' || undefined, 'updateUser': 'updateUser' || undefined }) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.visible = false this.$emit('refreshDataList') } }) } else { this.$message.error(data.msg) } }) } }) } } }</script>/\*\* \* @description fitness\_plans \* @author 大大的盆 \*/@RestController@RequestMapping(value = "/fitnessPlans")public class FitnessPlansController { @Resource private FitnessPlansService fitnessPlansService; /\*\* \* 新增 \* @author 大大的盆 \*\*/ @RequestMapping("/insert") public Object insert(FitnessPlans fitnessPlans){ return fitnessPlansService.insert(fitnessPlans); } /\*\* \* 刪除 \* @author 大大的盆 \*\*/ @RequestMapping("/delete") public Object delete(int id){ return fitnessPlansService.delete(id); } /\*\* \* 更新 \* @author 大大的盆 \*\*/ @RequestMapping("/update") public Object update(FitnessPlans fitnessPlans){ return fitnessPlansService.update(fitnessPlans); } /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ @RequestMapping("/load") public Object load(int id){ return fitnessPlansService.load(id); } /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ @RequestMapping("/pageList") public Map<String, Object> pageList(@RequestParam(required = false, defaultValue = "0") int offset, @RequestParam(required = false, defaultValue = "10") int pagesize) { return fitnessPlansService.pageList(offset, pagesize); }}/\*\* \* @description fitness\_plans \* @author 大大的盆 \*/public interface FitnessPlansService { /\*\* \* 新增 \*/ public Object insert(FitnessPlans fitnessPlans); /\*\* \* 删除 \*/ public Object delete(int id); /\*\* \* 更新 \*/ public Object update(FitnessPlans fitnessPlans); /\*\* \* 根据主键 id 查询 \*/ public FitnessPlans load(int id); /\*\* \* 分页查询 \*/ public Map<String,Object> pageList(int offset, int pagesize);}/\*\* \* @description fitness\_plans \* @author 大大的盆 \*/@Servicepublic class FitnessPlansServiceImpl implements FitnessPlansService { @Resource private FitnessPlansMapper fitnessPlansMapper; @Override public Object insert(FitnessPlans fitnessPlans) { // valid if (fitnessPlans == null) { return ReturnT.error("必要参数缺失"); } fitnessPlansMapper.insert(fitnessPlans); return ReturnT.success(); } @Override public Object delete(int id) { int ret = fitnessPlansMapper.delete(id); return ret>0?ReturnT.success():ReturnT.error(); } @Override public Object update(FitnessPlans fitnessPlans) { int ret = fitnessPlansMapper.update(fitnessPlans); return ret>0?ReturnT.success():ReturnT.error(); } @Override public FitnessPlans load(int id) { return fitnessPlansMapper.load(id); } @Override public Map<String,Object> pageList(int offset, int pagesize) { List<FitnessPlans> pageList = fitnessPlansMapper.pageList(offset, pagesize); int totalCount = fitnessPlansMapper.pageListCount(offset, pagesize); // result Map<String, Object> result = new HashMap<String, Object>(); result.put("pageList", pageList); result.put("totalCount", totalCount); return result; }}/\*\* \* @description fitness\_plans \* @author 大大的盆 \*/@Mapper@Repositorypublic interface FitnessPlansMapper { /\*\* \* 新增 \* @author 大大的盆 \*\*/ int insert(FitnessPlans fitnessPlans); /\*\* \* 刪除 \* @author 大大的盆 \*\*/ int delete(int id); /\*\* \* 更新 \* @author 大大的盆 \*\*/ int update(FitnessPlans fitnessPlans); /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ FitnessPlans load(int id); /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ List<FitnessPlans> pageList(int offset,int pagesize); /\*\* \* 查询 分页查询 count \* @author 大大的盆 \*\*/ int pageListCount(int offset,int pagesize);}<?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="HBJZGCXY.com.dao.FitnessPlansMapper"> <resultMap id="BaseResultMap" type="HBJZGCXY.com.entity.FitnessPlans" > <result column="user\_id" property="userId" /> <result column="plan\_details" property="planDetails" /> <result column="start\_date" property="startDate" /> <result column="end\_date" property="endDate" /> <result column="create\_time" property="createTime" /> <result column="create\_user" property="createUser" /> <result column="update\_time" property="updateTime" /> <result column="update\_user" property="updateUser" /> </resultMap> <sql id="Base\_Column\_List"> user\_id, plan\_details, start\_date, end\_date, create\_time, create\_user, update\_time, update\_user </sql> <insert id="insert" useGeneratedKeys="true" keyColumn="id" keyProperty="id" parameterType="HBJZGCXY.com.entity.FitnessPlans"> INSERT INTO fitness\_plans <trim prefix="(" suffix=")" suffixOverrides=","> <if test="null != userId and '' != userId"> user\_id, </if> <if test="null != planDetails and '' != planDetails"> plan\_details, </if> <if test="null != startDate and '' != startDate"> start\_date, </if> <if test="null != endDate and '' != endDate"> end\_date, </if> <if test="null != createTime and '' != createTime"> create\_time, </if> <if test="null != createUser and '' != createUser"> create\_user, </if> <if test="null != updateTime and '' != updateTime"> update\_time, </if> <if test="null != updateUser and '' != updateUser"> update\_user </if> </trim> <trim prefix="values (" suffix=")" suffixOverrides=","> <if test="null != userId and '' != userId"> #{userId}, </if> <if test="null != planDetails and '' != planDetails"> #{planDetails}, </if> <if test="null != startDate and '' != startDate"> #{startDate}, </if> <if test="null != endDate and '' != endDate"> #{endDate}, </if> <if test="null != createTime and '' != createTime"> #{createTime}, </if> <if test="null != createUser and '' != createUser"> #{createUser}, </if> <if test="null != updateTime and '' != updateTime"> #{updateTime}, </if> <if test="null != updateUser and '' != updateUser"> #{updateUser} </if> </trim> </insert> <delete id="delete" > DELETE FROM fitness\_plans WHERE id = #{id} </delete> <update id="update" parameterType="HBJZGCXY.com.entity.FitnessPlans"> UPDATE fitness\_plans <set> <if test="null != userId and '' != userId">user\_id = #{userId},</if> <if test="null != planDetails and '' != planDetails">plan\_details = #{planDetails},</if> <if test="null != startDate and '' != startDate">start\_date = #{startDate},</if> <if test="null != endDate and '' != endDate">end\_date = #{endDate},</if> <if test="null != createTime and '' != createTime">create\_time = #{createTime},</if> <if test="null != createUser and '' != createUser">create\_user = #{createUser},</if> <if test="null != updateTime and '' != updateTime">update\_time = #{updateTime},</if> <if test="null != updateUser and '' != updateUser">update\_user = #{updateUser}</if> </set> WHERE id = #{id} </update> <select id="load" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM fitness\_plans WHERE id = #{id} </select> <select id="pageList" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM fitness\_plans LIMIT #{offset}, #{pageSize} </select> <select id="pageListCount" resultType="java.lang.Integer"> SELECT count(1) FROM fitness\_plans </select></mapper>/\*\* \* @description fitness\_plans \* @author 大大的盆 \*/public class FitnessPlans implements Serializable { private static final long serialVersionUID = 1L; private String userId; private String planDetails; private String startDate; private String endDate; private String createTime; private String createUser; private String updateTime; private String updateUser; public FitnessPlans() { } public String getUserId() { return userId; } public void setUserId(String userId) { this.userId = userId; } public String getPlanDetails() { return planDetails; } public void setPlanDetails(String planDetails) { this.planDetails = planDetails; } public String getStartDate() { return startDate; } public void setStartDate(String startDate) { this.startDate = startDate; } public String getEndDate() { return endDate; } public void setEndDate(String endDate) { this.endDate = endDate; } public String getCreateTime() { return createTime; } public void setCreateTime(String createTime) { this.createTime = createTime; } public String getCreateUser() { return createUser; } public void setCreateUser(String createUser) { this.createUser = createUser; } public String getUpdateTime() { return updateTime; } public void setUpdateTime(String updateTime) { this.updateTime = updateTime; } public String getUpdateUser() { return updateUser; } public void setUpdateUser(String updateUser) { this.updateUser = updateUser; }}<template> <div class="mod-config"> <el-form :inline="true" :model="dataForm" @keyup.enter.native="getDataList()"> <el-form-item> <el-input v-model="dataForm.key" placeholder="参数名" clearable></el-input> </el-form-item> <el-form-item> <el-button @click="getDataList()">查询</el-button> <el-button v-if="isAuth('generator:fitnessPlans:save')" type="primary" @click="addOrUpdateHandle()">新增</el-button> <el-button v-if="isAuth('generator:fitnessPlans:delete')" type="danger" @click="deleteHandle()" :disabled="dataListSelections.length <= 0">批量删除</el-button> </el-form-item> </el-form> <el-table :data="dataList" border v-loading="dataListLoading" @selection-change="selectionChangeHandle" style="width: 100%;"> <el-table-column type="selection" header-align="center" align="center" width="50"> </el-table-column> <el-table-column prop="userId" header-align="center" align="center" label="user\_id"> </el-table-column> <el-table-column prop="planDetails" header-align="center" align="center" label="plan\_details"> </el-table-column> <el-table-column prop="startDate" header-align="center" align="center" label="start\_date"> </el-table-column> <el-table-column prop="endDate" header-align="center" align="center" label="end\_date"> </el-table-column> <el-table-column prop="createTime" header-align="center" align="center" label="create\_time"> </el-table-column> <el-table-column prop="createUser" header-align="center" align="center" label="create\_user"> </el-table-column> <el-table-column prop="updateTime" header-align="center" align="center" label="update\_time"> </el-table-column> <el-table-column prop="updateUser" header-align="center" align="center" label="update\_user"> </el-table-column> <el-table-column fixed="right" header-align="center" align="center" width="150" label="操作"> <template slot-scope="scope"> <el-button type="text" size="small" @click="addOrUpdateHandle(scope.row.fitnessPlansId)">修改</el-button> <el-button type="text" size="small" @click="deleteHandle(scope.row.fitnessPlansId)">删除</el-button> </template> </el-table-column> </el-table> <el-pagination @size-change="sizeChangeHandle" @current-change="currentChangeHandle" :current-page="pageIndex" :page-sizes="[10, 20, 50, 100]" :page-size="pageSize" :total="totalPage" layout="total, sizes, prev, pager, next, jumper"> </el-pagination> <!-- 弹窗, 新增 / 修改 --> <add-or-update v-if="addOrUpdateVisible" ref="addOrUpdate" @refreshDataList="getDataList"></add-or-update> </div></template><script> import AddOrUpdate from './fitnessPlans-add-or-update' export default { data () { return { dataForm: { key: '' }, dataList: [], pageIndex: 1, pageSize: 10, totalPage: 0, dataListLoading: false, dataListSelections: [], addOrUpdateVisible: false } }, components: { AddOrUpdate }, activated () { this.getDataList() }, methods: { // 获取数据列表 getDataList () { this.dataListLoading = true this.$http({ url: this.$http.adornUrl('/generator/fitnessPlans/list'), method: 'get', params: this.$http.adornParams({ 'page': this.pageIndex, 'limit': this.pageSize, 'key': this.dataForm.key }) }).then(({data}) => { if (data && data.code === 0) { this.dataList = data.page.list this.totalPage = data.page.totalCount } else { this.dataList = [] this.totalPage = 0 } this.dataListLoading = false }) }, // 每页数 sizeChangeHandle (val) { this.pageSize = val this.pageIndex = 1 this.getDataList() }, // 当前页 currentChangeHandle (val) { this.pageIndex = val this.getDataList() }, // 多选 selectionChangeHandle (val) { this.dataListSelections = val }, // 新增 / 修改 addOrUpdateHandle (id) { this.addOrUpdateVisible = true this.$nextTick(() => { this.$refs.addOrUpdate.init(id) }) }, // 删除 deleteHandle (id) { var ids = id ? [id] : this.dataListSelections.map(item => { return item.fitnessPlansId }) this.$confirm(`确定对[id=${ids.join(',')}]进行[${id ? '删除' : '批量删除'}]操作?`, '提示', { confirmButtonText: '确定', cancelButtonText: '取消', type: 'warning' }).then(() => { this.$http({ url: this.$http.adornUrl('/generator/fitnessPlans/delete'), method: 'post', data: this.$http.adornData(ids, false) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.getDataList() } }) } else { this.$message.error(data.msg) } }) }) } } }</script><template> <el-dialog :title="!dataForm.fitnessPlansId ? '新增' : '修改'" :close-on-click-modal="false" :visible.sync="visible"> <el-form :model="dataForm" :rules="dataRule" ref="dataForm" @keyup.enter.native="dataFormSubmit()" label-width="80px"> <el-form-item label="user\_id" prop="userId"> <el-input v-model="dataForm.userId" placeholder="user\_id"></el-input> </el-form-item> <el-form-item label="plan\_details" prop="planDetails"> <el-input v-model="dataForm.planDetails" placeholder="plan\_details"></el-input> </el-form-item> <el-form-item label="start\_date" prop="startDate"> <el-input v-model="dataForm.startDate" placeholder="start\_date"></el-input> </el-form-item> <el-form-item label="end\_date" prop="endDate"> <el-input v-model="dataForm.endDate" placeholder="end\_date"></el-input> </el-form-item> <el-form-item label="create\_time" prop="createTime"> <el-input v-model="dataForm.createTime" placeholder="create\_time"></el-input> </el-form-item> <el-form-item label="create\_user" prop="createUser"> <el-input v-model="dataForm.createUser" placeholder="create\_user"></el-input> </el-form-item> <el-form-item label="update\_time" prop="updateTime"> <el-input v-model="dataForm.updateTime" placeholder="update\_time"></el-input> </el-form-item> <el-form-item label="update\_user" prop="updateUser"> <el-input v-model="dataForm.updateUser" placeholder="update\_user"></el-input> </el-form-item> </el-form> <span slot="footer" class="dialog-footer"> <el-button @click="visible = false">取消</el-button> <el-button type="primary" @click="dataFormSubmit()">确定</el-button> </span> </el-dialog></template><script> export default { data () { return { visible: false, dataForm: { userId: '', planDetails: '', startDate: '', endDate: '', createTime: '', createUser: '', updateTime: '', updateUser: '' }, dataRule: { userId: [{ required: true, message: 'user\_id不能为空', trigger: 'blur' }], planDetails: [{ required: true, message: 'plan\_details不能为空', trigger: 'blur' }], startDate: [{ required: true, message: 'start\_date不能为空', trigger: 'blur' }], endDate: [{ required: true, message: 'end\_date不能为空', trigger: 'blur' }], createTime: [{ required: true, message: 'create\_time不能为空', trigger: 'blur' }], createUser: [{ required: true, message: 'create\_user不能为空', trigger: 'blur' }], updateTime: [{ required: true, message: 'update\_time不能为空', trigger: 'blur' }], updateUser: [{ required: true, message: 'update\_user不能为空', trigger: 'blur' }] } } }, methods: { init (id) { this.dataForm.fitnessPlansId = id || 0 this.visible = true this.$nextTick(() => { this.$refs['dataForm'].resetFields() if (this.dataForm.fitnessPlansId) { this.$http({ url: this.$http.adornUrl(`/generator/fitnessPlans/info/${this.dataForm.fitnessPlansId}`), method: 'get', params: this.$http.adornParams() }).then(({data}) => { if (data && data.code === 0) { this.dataForm.userId = data.fitnessPlans.userId this.dataForm.planDetails = data.fitnessPlans.planDetails this.dataForm.startDate = data.fitnessPlans.startDate this.dataForm.endDate = data.fitnessPlans.endDate this.dataForm.createTime = data.fitnessPlans.createTime this.dataForm.createUser = data.fitnessPlans.createUser this.dataForm.updateTime = data.fitnessPlans.updateTime this.dataForm.updateUser = data.fitnessPlans.updateUser } }) } }) }, // 表单提交 dataFormSubmit () { this.$refs['dataForm'].validate((valid) => { if (valid) { this.$http({ url: this.$http.adornUrl(`/generator/fitnessPlans/${this.dataForm.fitnessPlansId? 'save' : 'update'}`), method: 'post', data: this.$http.adornData({ 'userId': 'userId' || undefined, 'planDetails': 'planDetails' || undefined, 'startDate': 'startDate' || undefined, 'endDate': 'endDate' || undefined, 'createTime': 'createTime' || undefined, 'createUser': 'createUser' || undefined, 'updateTime': 'updateTime' || undefined, 'updateUser': 'updateUser' || undefined }) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.visible = false this.$emit('refreshDataList') } }) } else { this.$message.error(data.msg) } }) } }) } } }</script>/\*\* \* @description payment\_records \* @author 大大的盆 \*/@RestController@RequestMapping(value = "/paymentRecords")public class PaymentRecordsController { @Resource private PaymentRecordsService paymentRecordsService; /\*\* \* 新增 \* @author 大大的盆 \*\*/ @RequestMapping("/insert") public Object insert(PaymentRecords paymentRecords){ return paymentRecordsService.insert(paymentRecords); } /\*\* \* 刪除 \* @author 大大的盆 \*\*/ @RequestMapping("/delete") public Object delete(int id){ return paymentRecordsService.delete(id); } /\*\* \* 更新 \* @author 大大的盆 \*\*/ @RequestMapping("/update") public Object update(PaymentRecords paymentRecords){ return paymentRecordsService.update(paymentRecords); } /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ @RequestMapping("/load") public Object load(int id){ return paymentRecordsService.load(id); } /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ @RequestMapping("/pageList") public Map<String, Object> pageList(@RequestParam(required = false, defaultValue = "0") int offset, @RequestParam(required = false, defaultValue = "10") int pagesize) { return paymentRecordsService.pageList(offset, pagesize); }}/\*\* \* @description payment\_records \* @author 大大的盆 \*/public interface PaymentRecordsService { /\*\* \* 新增 \*/ public Object insert(PaymentRecords paymentRecords); /\*\* \* 删除 \*/ public Object delete(int id); /\*\* \* 更新 \*/ public Object update(PaymentRecords paymentRecords); /\*\* \* 根据主键 id 查询 \*/ public PaymentRecords load(int id); /\*\* \* 分页查询 \*/ public Map<String,Object> pageList(int offset, int pagesize);}/\*\* \* @description payment\_records \* @author 大大的盆 \*/@Servicepublic class PaymentRecordsServiceImpl implements PaymentRecordsService { @Resource private PaymentRecordsMapper paymentRecordsMapper; @Override public Object insert(PaymentRecords paymentRecords) { // valid if (paymentRecords == null) { return ReturnT.error("必要参数缺失"); } paymentRecordsMapper.insert(paymentRecords); return ReturnT.success(); } @Override public Object delete(int id) { int ret = paymentRecordsMapper.delete(id); return ret>0?ReturnT.success():ReturnT.error(); } @Override public Object update(PaymentRecords paymentRecords) { int ret = paymentRecordsMapper.update(paymentRecords); return ret>0?ReturnT.success():ReturnT.error(); } @Override public PaymentRecords load(int id) { return paymentRecordsMapper.load(id); } @Override public Map<String,Object> pageList(int offset, int pagesize) { List<PaymentRecords> pageList = paymentRecordsMapper.pageList(offset, pagesize); int totalCount = paymentRecordsMapper.pageListCount(offset, pagesize); // result Map<String, Object> result = new HashMap<String, Object>(); result.put("pageList", pageList); result.put("totalCount", totalCount); return result; }}/\*\* \* @description payment\_records \* @author 大大的盆 \*/@Mapper@Repositorypublic interface PaymentRecordsMapper { /\*\* \* 新增 \* @author 大大的盆 \*\*/ int insert(PaymentRecords paymentRecords); /\*\* \* 刪除 \* @author 大大的盆 \*\*/ int delete(int id); /\*\* \* 更新 \* @author 大大的盆 \*\*/ int update(PaymentRecords paymentRecords); /\*\* \* 查询 根据主键 id 查询 \* @author 大大的盆 \*\*/ PaymentRecords load(int id); /\*\* \* 查询 分页查询 \* @author 大大的盆 \*\*/ List<PaymentRecords> pageList(int offset,int pagesize); /\*\* \* 查询 分页查询 count \* @author 大大的盆 \*\*/ int pageListCount(int offset,int pagesize);}<?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="HBJZGCXY.com.dao.PaymentRecordsMapper"> <resultMap id="BaseResultMap" type="HBJZGCXY.com.entity.PaymentRecords" > <result column="user\_id" property="userId" /> <result column="amount" property="amount" /> <result column="payment\_date" property="paymentDate" /> <result column="payment\_method" property="paymentMethod" /> <result column="create\_time" property="createTime" /> <result column="create\_user" property="createUser" /> <result column="update\_time" property="updateTime" /> <result column="update\_user" property="updateUser" /> </resultMap> <sql id="Base\_Column\_List"> user\_id, amount, payment\_date, payment\_method, create\_time, create\_user, update\_time, update\_user </sql> <insert id="insert" useGeneratedKeys="true" keyColumn="id" keyProperty="id" parameterType="HBJZGCXY.com.entity.PaymentRecords"> INSERT INTO payment\_records <trim prefix="(" suffix=")" suffixOverrides=","> <if test="null != userId and '' != userId"> user\_id, </if> <if test="null != amount and '' != amount"> amount, </if> <if test="null != paymentDate and '' != paymentDate"> payment\_date, </if> <if test="null != paymentMethod and '' != paymentMethod"> payment\_method, </if> <if test="null != createTime and '' != createTime"> create\_time, </if> <if test="null != createUser and '' != createUser"> create\_user, </if> <if test="null != updateTime and '' != updateTime"> update\_time, </if> <if test="null != updateUser and '' != updateUser"> update\_user </if> </trim> <trim prefix="values (" suffix=")" suffixOverrides=","> <if test="null != userId and '' != userId"> #{userId}, </if> <if test="null != amount and '' != amount"> #{amount}, </if> <if test="null != paymentDate and '' != paymentDate"> #{paymentDate}, </if> <if test="null != paymentMethod and '' != paymentMethod"> #{paymentMethod}, </if> <if test="null != createTime and '' != createTime"> #{createTime}, </if> <if test="null != createUser and '' != createUser"> #{createUser}, </if> <if test="null != updateTime and '' != updateTime"> #{updateTime}, </if> <if test="null != updateUser and '' != updateUser"> #{updateUser} </if> </trim> </insert> <delete id="delete" > DELETE FROM payment\_records WHERE id = #{id} </delete> <update id="update" parameterType="HBJZGCXY.com.entity.PaymentRecords"> UPDATE payment\_records <set> <if test="null != userId and '' != userId">user\_id = #{userId},</if> <if test="null != amount and '' != amount">amount = #{amount},</if> <if test="null != paymentDate and '' != paymentDate">payment\_date = #{paymentDate},</if> <if test="null != paymentMethod and '' != paymentMethod">payment\_method = #{paymentMethod},</if> <if test="null != createTime and '' != createTime">create\_time = #{createTime},</if> <if test="null != createUser and '' != createUser">create\_user = #{createUser},</if> <if test="null != updateTime and '' != updateTime">update\_time = #{updateTime},</if> <if test="null != updateUser and '' != updateUser">update\_user = #{updateUser}</if> </set> WHERE id = #{id} </update> <select id="load" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM payment\_records WHERE id = #{id} </select> <select id="pageList" resultMap="BaseResultMap"> SELECT <include refid="Base\_Column\_List" /> FROM payment\_records LIMIT #{offset}, #{pageSize} </select> <select id="pageListCount" resultType="java.lang.Integer"> SELECT count(1) FROM payment\_records </select></mapper>/\*\* \* @description payment\_records \* @author 大大的盆 \*/public class PaymentRecords implements Serializable { private static final long serialVersionUID = 1L; private String userId; private String amount; private String paymentDate; private String paymentMethod; private String createTime; private String createUser; private String updateTime; private String updateUser; public PaymentRecords() { } public String getUserId() { return userId; } public void setUserId(String userId) { this.userId = userId; } public String getAmount() { return amount; } public void setAmount(String amount) { this.amount = amount; } public String getPaymentDate() { return paymentDate; } public void setPaymentDate(String paymentDate) { this.paymentDate = paymentDate; } public String getPaymentMethod() { return paymentMethod; } public void setPaymentMethod(String paymentMethod) { this.paymentMethod = paymentMethod; } public String getCreateTime() { return createTime; } public void setCreateTime(String createTime) { this.createTime = createTime; } public String getCreateUser() { return createUser; } public void setCreateUser(String createUser) { this.createUser = createUser; } public String getUpdateTime() { return updateTime; } public void setUpdateTime(String updateTime) { this.updateTime = updateTime; } public String getUpdateUser() { return updateUser; } public void setUpdateUser(String updateUser) { this.updateUser = updateUser; }}<template> <div class="mod-config"> <el-form :inline="true" :model="dataForm" @keyup.enter.native="getDataList()"> <el-form-item> <el-input v-model="dataForm.key" placeholder="参数名" clearable></el-input> </el-form-item> <el-form-item> <el-button @click="getDataList()">查询</el-button> <el-button v-if="isAuth('generator:paymentRecords:save')" type="primary" @click="addOrUpdateHandle()">新增</el-button> <el-button v-if="isAuth('generator:paymentRecords:delete')" type="danger" @click="deleteHandle()" :disabled="dataListSelections.length <= 0">批量删除</el-button> </el-form-item> </el-form> <el-table :data="dataList" border v-loading="dataListLoading" @selection-change="selectionChangeHandle" style="width: 100%;"> <el-table-column type="selection" header-align="center" align="center" width="50"> </el-table-column> <el-table-column prop="userId" header-align="center" align="center" label="user\_id"> </el-table-column> <el-table-column prop="amount" header-align="center" align="center" label="amount"> </el-table-column> <el-table-column prop="paymentDate" header-align="center" align="center" label="payment\_date"> </el-table-column> <el-table-column prop="paymentMethod" header-align="center" align="center" label="payment\_method"> </el-table-column> <el-table-column prop="createTime" header-align="center" align="center" label="create\_time"> </el-table-column> <el-table-column prop="createUser" header-align="center" align="center" label="create\_user"> </el-table-column> <el-table-column prop="updateTime" header-align="center" align="center" label="update\_time"> </el-table-column> <el-table-column prop="updateUser" header-align="center" align="center" label="update\_user"> </el-table-column> <el-table-column fixed="right" header-align="center" align="center" width="150" label="操作"> <template slot-scope="scope"> <el-button type="text" size="small" @click="addOrUpdateHandle(scope.row.paymentRecordsId)">修改</el-button> <el-button type="text" size="small" @click="deleteHandle(scope.row.paymentRecordsId)">删除</el-button> </template> </el-table-column> </el-table> <el-pagination @size-change="sizeChangeHandle" @current-change="currentChangeHandle" :current-page="pageIndex" :page-sizes="[10, 20, 50, 100]" :page-size="pageSize" :total="totalPage" layout="total, sizes, prev, pager, next, jumper"> </el-pagination> <!-- 弹窗, 新增 / 修改 --> <add-or-update v-if="addOrUpdateVisible" ref="addOrUpdate" @refreshDataList="getDataList"></add-or-update> </div></template><script> import AddOrUpdate from './paymentRecords-add-or-update' export default { data () { return { dataForm: { key: '' }, dataList: [], pageIndex: 1, pageSize: 10, totalPage: 0, dataListLoading: false, dataListSelections: [], addOrUpdateVisible: false } }, components: { AddOrUpdate }, activated () { this.getDataList() }, methods: { // 获取数据列表 getDataList () { this.dataListLoading = true this.$http({ url: this.$http.adornUrl('/generator/paymentRecords/list'), method: 'get', params: this.$http.adornParams({ 'page': this.pageIndex, 'limit': this.pageSize, 'key': this.dataForm.key }) }).then(({data}) => { if (data && data.code === 0) { this.dataList = data.page.list this.totalPage = data.page.totalCount } else { this.dataList = [] this.totalPage = 0 } this.dataListLoading = false }) }, // 每页数 sizeChangeHandle (val) { this.pageSize = val this.pageIndex = 1 this.getDataList() }, // 当前页 currentChangeHandle (val) { this.pageIndex = val this.getDataList() }, // 多选 selectionChangeHandle (val) { this.dataListSelections = val }, // 新增 / 修改 addOrUpdateHandle (id) { this.addOrUpdateVisible = true this.$nextTick(() => { this.$refs.addOrUpdate.init(id) }) }, // 删除 deleteHandle (id) { var ids = id ? [id] : this.dataListSelections.map(item => { return item.paymentRecordsId }) this.$confirm(`确定对[id=${ids.join(',')}]进行[${id ? '删除' : '批量删除'}]操作?`, '提示', { confirmButtonText: '确定', cancelButtonText: '取消', type: 'warning' }).then(() => { this.$http({ url: this.$http.adornUrl('/generator/paymentRecords/delete'), method: 'post', data: this.$http.adornData(ids, false) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.getDataList() } }) } else { this.$message.error(data.msg) } }) }) } } }</script><template> <el-dialog :title="!dataForm.paymentRecordsId ? '新增' : '修改'" :close-on-click-modal="false" :visible.sync="visible"> <el-form :model="dataForm" :rules="dataRule" ref="dataForm" @keyup.enter.native="dataFormSubmit()" label-width="80px"> <el-form-item label="user\_id" prop="userId"> <el-input v-model="dataForm.userId" placeholder="user\_id"></el-input> </el-form-item> <el-form-item label="amount" prop="amount"> <el-input v-model="dataForm.amount" placeholder="amount"></el-input> </el-form-item> <el-form-item label="payment\_date" prop="paymentDate"> <el-input v-model="dataForm.paymentDate" placeholder="payment\_date"></el-input> </el-form-item> <el-form-item label="payment\_method" prop="paymentMethod"> <el-input v-model="dataForm.paymentMethod" placeholder="payment\_method"></el-input> </el-form-item> <el-form-item label="create\_time" prop="createTime"> <el-input v-model="dataForm.createTime" placeholder="create\_time"></el-input> </el-form-item> <el-form-item label="create\_user" prop="createUser"> <el-input v-model="dataForm.createUser" placeholder="create\_user"></el-input> </el-form-item> <el-form-item label="update\_time" prop="updateTime"> <el-input v-model="dataForm.updateTime" placeholder="update\_time"></el-input> </el-form-item> <el-form-item label="update\_user" prop="updateUser"> <el-input v-model="dataForm.updateUser" placeholder="update\_user"></el-input> </el-form-item> </el-form> <span slot="footer" class="dialog-footer"> <el-button @click="visible = false">取消</el-button> <el-button type="primary" @click="dataFormSubmit()">确定</el-button> </span> </el-dialog></template><script> export default { data () { return { visible: false, dataForm: { userId: '', amount: '', paymentDate: '', paymentMethod: '', createTime: '', createUser: '', updateTime: '', updateUser: '' }, dataRule: { userId: [{ required: true, message: 'user\_id不能为空', trigger: 'blur' }], amount: [{ required: true, message: 'amount不能为空', trigger: 'blur' }], paymentDate: [{ required: true, message: 'payment\_date不能为空', trigger: 'blur' }], paymentMethod: [{ required: true, message: 'payment\_method不能为空', trigger: 'blur' }], createTime: [{ required: true, message: 'create\_time不能为空', trigger: 'blur' }], createUser: [{ required: true, message: 'create\_user不能为空', trigger: 'blur' }], updateTime: [{ required: true, message: 'update\_time不能为空', trigger: 'blur' }], updateUser: [{ required: true, message: 'update\_user不能为空', trigger: 'blur' }] } } }, methods: { init (id) { this.dataForm.paymentRecordsId = id || 0 this.visible = true this.$nextTick(() => { this.$refs['dataForm'].resetFields() if (this.dataForm.paymentRecordsId) { this.$http({ url: this.$http.adornUrl(`/generator/paymentRecords/info/${this.dataForm.paymentRecordsId}`), method: 'get', params: this.$http.adornParams() }).then(({data}) => { if (data && data.code === 0) { this.dataForm.userId = data.paymentRecords.userId this.dataForm.amount = data.paymentRecords.amount this.dataForm.paymentDate = data.paymentRecords.paymentDate this.dataForm.paymentMethod = data.paymentRecords.paymentMethod this.dataForm.createTime = data.paymentRecords.createTime this.dataForm.createUser = data.paymentRecords.createUser this.dataForm.updateTime = data.paymentRecords.updateTime this.dataForm.updateUser = data.paymentRecords.updateUser } }) } }) }, // 表单提交 dataFormSubmit () { this.$refs['dataForm'].validate((valid) => { if (valid) { this.$http({ url: this.$http.adornUrl(`/generator/paymentRecords/${this.dataForm.paymentRecordsId? 'save' : 'update'}`), method: 'post', data: this.$http.adornData({ 'userId': 'userId' || undefined, 'amount': 'amount' || undefined, 'paymentDate': 'paymentDate' || undefined, 'paymentMethod': 'paymentMethod' || undefined, 'createTime': 'createTime' || undefined, 'createUser': 'createUser' || undefined, 'updateTime': 'updateTime' || undefined, 'updateUser': 'updateUser' || undefined }) }).then(({data}) => { if (data && data.code === 0) { this.$message({ message: '操作成功', type: 'success', duration: 1500, onClose: () => { this.visible = false this.$emit('refreshDataList') } }) } else { this.$message.error(data.msg) } }) } }) } } }</script>