这一节配置了菜单管理树形表格里面的icon样式，以及角色管理里分配菜单按钮的实现。总的来说不是很难，难的在下一节动态路由的设置。

# 代码

## Menu.vue

|  |
| --- |
| <template>  <div>   <div style="padding: 10px 0">  <el-input style="width: 200px" placeholder="请输入名称" suffix-icon="el-icon-search" v-model="name"></el-input> <!-- <el-input style="width: 200px" placeholder="请输入邮箱" suffix-icon="el-icon-message" class="ml-5" v-model="email"></el-input>--> <!-- <el-input style="width: 200px" placeholder="请输入地址" suffix-icon="el-icon-position" class="ml-5" v-model="address"></el-input>-->  <el-button class="ml-5" type="primary" @click="load">搜索</el-button>  <el-button type="warning" @click="reset">重置</el-button>  </div>   <div style="margin: 10px 0">  <el-button type="primary" @click="handleAdd()">新增 <i class="el-icon-circle-plus-outline"></i></el-button>  <el-popconfirm  class="ml-5"  confirm-button-text='确定'  cancel-button-text='我再想想'  icon="el-icon-info"  icon-color="red"  title="您确定批量删除这些数据吗？"  @confirm="delBatch"  >  <el-button type="danger" slot="reference">批量删除 <i class="el-icon-remove-outline"></i></el-button>  </el-popconfirm> <!-- <el-upload action="http://localhost:9090/user/import" :show-file-list="false" accept="xlsx" :on-success="handleExcelImportSuccess" style="display:inline-block">--> <!-- <el-button type="primary" class="ml-5">导入 <i class="el-icon-bottom"></i></el-button>--> <!-- </el-upload>--> <!-- <el-button type="primary" @click="exp" class="ml-5">导出 <i class="el-icon-top"></i></el-button>-->  </div>   <el-table :data="tableData" border stripe :header-cell-class-name="'headerBg'"  row-key="id" default-expand-all @selection-change="handleSelectionChange">  <el-table-column type="selection" width="55"></el-table-column>  <el-table-column prop="id" label="ID" width="80px"></el-table-column>  <el-table-column prop="name" label="名称"></el-table-column>  <el-table-column prop="path" label="路径"></el-table-column>  <el-table-column label="图标" align="center">  <template slot-scope="scope">  <i :class="scope.row.icon" style="font-size: 20px" />  </template>  </el-table-column>  <el-table-column prop="description" label="描述"></el-table-column>   <el-table-column label="操作" width="300px" align="center">  <template slot-scope="scope">  <el-button type="primary" @click="handleAdd(scope.row.id)" v-if="!scope.row.pid && !scope.row.path">新增子菜单 <i class="el-icon-plus"></i></el-button>  <el-button type="success" @click="handleEdit(scope.row)">编辑 <i class="el-icon-edit"></i></el-button>  <el-popconfirm  class="ml-5"  confirm-button-text='确定'  cancel-button-text='我再想想'  icon="el-icon-info"  icon-color="red"  title="您确定删除吗？"  @confirm="del(scope.row.id)"  >  <el-button type="danger" slot="reference">删除 <i class="el-icon-remove-outline"></i></el-button>  </el-popconfirm>  </template>  </el-table-column>  </el-table>   <el-dialog title="菜单信息" :visible.sync="dialogFormVisible" width="30%">  <el-form label-width="80px" size="small">  <el-form-item label="名称">  <el-input v-model="form.name" autocomplete="off"></el-input>  </el-form-item>  <el-form-item label="路径">  <el-input v-model="form.path" autocomplete="off"></el-input>  </el-form-item>  <el-form-item label="图标">  <el-select clearable v-model="form.icon" placeholder="请选择" style="width: 100%">  <el-option v-for="item in options" :key="item.name" :label="item.name" :value="item.value">  <i :class="item.value"/> {{ item.name}}  </el-option>  </el-select>  </el-form-item>  <el-form-item label="描述">  <el-input v-model="form.description" autocomplete="off"></el-input>  </el-form-item>  </el-form>  <div slot="footer" class="dialog-footer">  <el-button @click="dialogFormVisible = false">取 消</el-button>  <el-button type="primary" @click="save">确 定</el-button>  </div>  </el-dialog>  </div> </template>  <script> export default {  name: "User",  data(){  return {  tableData: [],  total: 0,  pageNum:1,  pageSize:10,  name:"",  form:{},  dialogFormVisible:false,  multipleSelection:[],  options:[]  }  },  created() {  this.load()  },  methods:{  load(){  //与下面注释的代码对比，我改动的是api的替换和赋值方式的替换  this.request.get("/menu",{  params:{  name:this.name,  }  }).then(res =>{  this.tableData=res.data  })  // fetch("http://localhost:9090/user/page?pageNum="+this.pageNum+"&pageSize="+this.pageSize+"&username="+this.username)  // .then(res => res.json()).then(res =>{  // console.log(res)  // //数据绑定  // this.tableData = res.data  // this.total=res.total  // })  },  save(){  this.request.post("/menu",this.form).then(res =>{  if(res.code ==='200'){  this.$message.success("保存成功")  this.dialogFormVisible=false  this.load()  }  else {  this.$message.error("保存失败")  }  })  },  handleAdd(pid){  this.dialogFormVisible=true  this.form={}  if(pid){  this.form.pid = pid  }  },  handleEdit(row){  //this.form=row //赋予表单数据,这样有个bug就是，在点击取消的时候表单上仍然有值，不过数据库里面的值没有变化,  this.form=***JSON***.parse(***JSON***.stringify(row)); //这是解决办法  this.dialogFormVisible=true //打开对话框   //请求图标的数据  this.request.get("/menu/icons").then(res =>{  this.options=res.data  })  },  del(id){  this.request.delete("/menu/"+id).then(res =>{  if(res.code ==='200'){  this.$message.success("删除成功")  this.load()  }  else {  this.$message.error("删除失败")  }  })  },  handleSelectionChange(val){  ***console***.log(val)  this.multipleSelection=val //获取到的是个对象 不是我们后台需要的ids  },  delBatch(){  let ids = this.multipleSelection.map(v =>v.id) //[{},{},{}]=>[1,2,3]  this.request.post("/menu/del/batch",ids).then(res =>{  if(res.code ==='200'){  this.$message.success("批量删除成功")  this.load()  }  else {  this.$message.error("批量删除失败")  }  })  },  reset(){  this.name=""  this.load()  },  handleSizeChange(pageSize){  ***console***.log(pageSize)  this.pageSize=pageSize  this.load()  },  handleCurrentChange(pageNum){  ***console***.log(pageNum)  this.pageNum=pageNum  this.load()  },  exp(){  ***window***.open("http://localhost:9090/menu/export")  },  handleExcelImportSuccess(){  this.$message.success("导入成功")  this.load()  }  } } </script>  <style> .headerBg {  background: #eee**!important**; } </style> |

## Role.vue

|  |
| --- |
| <template>  <div>   <div style="padding: 10px 0">  <el-input style="width: 200px" placeholder="请输入名称" suffix-icon="el-icon-search" v-model="name"></el-input> <!-- <el-input style="width: 200px" placeholder="请输入邮箱" suffix-icon="el-icon-message" class="ml-5" v-model="email"></el-input>--> <!-- <el-input style="width: 200px" placeholder="请输入地址" suffix-icon="el-icon-position" class="ml-5" v-model="address"></el-input>-->  <el-button class="ml-5" type="primary" @click="load">搜索</el-button>  <el-button type="warning" @click="reset">重置</el-button>  </div>   <div style="margin: 10px 0">  <el-button type="primary" @click="handleAdd">新增 <i class="el-icon-circle-plus-outline"></i></el-button>  <el-popconfirm  class="ml-5"  confirm-button-text='确定'  cancel-button-text='我再想想'  icon="el-icon-info"  icon-color="red"  title="您确定批量删除这些数据吗？"  @confirm="delBatch"  >  <el-button type="danger" slot="reference">批量删除 <i class="el-icon-remove-outline"></i></el-button>  </el-popconfirm> <!-- <el-upload action="http://localhost:9090/user/import" :show-file-list="false" accept="xlsx" :on-success="handleExcelImportSuccess" style="display:inline-block">--> <!-- <el-button type="primary" class="ml-5">导入 <i class="el-icon-bottom"></i></el-button>--> <!-- </el-upload>--> <!-- <el-button type="primary" @click="exp" class="ml-5">导出 <i class="el-icon-top"></i></el-button>-->  </div>   <el-table :data="tableData" border stripe :header-cell-class-name="'headerBg'">  <el-table-column type="selection" width="55"></el-table-column>  <el-table-column prop="id" label="ID" width="80px"></el-table-column>  <el-table-column prop="name" label="名称"></el-table-column>  <el-table-column prop="description" label="描述"></el-table-column>   <el-table-column label="操作" width="280" align="center">  <template slot-scope="scope">  <el-button type="info" @click="selectMenu(scope.row.id)">分配菜单 <i class="el-icon-menu"></i></el-button>  <el-button type="success" @click="handleEdit(scope.row)">编辑 <i class="el-icon-edit"></i></el-button>  <el-popconfirm  class="ml-5"  confirm-button-text='确定'  cancel-button-text='我再想想'  icon="el-icon-info"  icon-color="red"  title="您确定删除吗？"  @confirm="del(scope.row.id)"  >  <el-button type="danger" slot="reference">删除 <i class="el-icon-remove-outline"></i></el-button>  </el-popconfirm>  </template>  </el-table-column>  </el-table>  <div style="padding: 10px 0">  <el-pagination  @size-change="handleSizeChange"  @current-change="handleCurrentChange"  :current-page="pageNum"  :page-sizes="[2, 5, 10, 20]"  :page-size="pageSize"  layout="total, sizes, prev, pager, next, jumper"  :total="total">  </el-pagination>  </div>   <el-dialog title="角色信息" :visible.sync="dialogFormVisible" width="30%">  <el-form label-width="80px" size="small">  <el-form-item label="名称">  <el-input v-model="form.name" autocomplete="off"></el-input>  </el-form-item>  <el-form-item label="描述">  <el-input v-model="form.description" autocomplete="off"></el-input>  </el-form-item>  </el-form>  <div slot="footer" class="dialog-footer">  <el-button @click="dialogFormVisible = false">取 消</el-button>  <el-button type="primary" @click="save">确 定</el-button>  </div>  </el-dialog>   <el-dialog title="菜单分配" :visible.sync="menuDialogVis" width="30%">  <el-tree  :props="props"  :data="menuData"  show-checkbox  node-key="id"  ref="tree"  :default-expanded-keys="expends"  :default-checked-keys="checks">  <span class="custom-tree-node" slot-scope="{ node, data }">  <span><i :class="data.icon"/>{{ data.name }}</span>  </span>  </el-tree>  <div slot="footer" class="dialog-footer">  <el-button @click="menuDialogVis = false">取 消</el-button>  <el-button type="primary" @click="saveRoleMenu">确 定</el-button>  </div>  </el-dialog>   </div> </template>  <script> export default {  name: "User",  data(){  return {  tableData: [],  total: 0,  pageNum:1,  pageSize:10,  name:"",  form:{},  dialogFormVisible:false,  menuDialogVis:false,  multipleSelection:[],  menuData:[],  props: {  label: 'name',  },  expends:[],  checks:[],  roleId:0  }  },  created() {  this.load()  },  methods:{  load(){  //与下面注释的代码对比，我改动的是api的替换和赋值方式的替换  this.request.get("/role/page",{  params:{  pageNum:this.pageNum,  pageSize:this.pageSize,  name:this.name,  }  }).then(res =>{  this.tableData=res.data.records  this.total=res.data.total  })  // fetch("http://localhost:9090/user/page?pageNum="+this.pageNum+"&pageSize="+this.pageSize+"&username="+this.username)  // .then(res => res.json()).then(res =>{  // console.log(res)  // //数据绑定  // this.tableData = res.data  // this.total=res.total  // })   //请求菜单数据   },  save(){  this.request.post("/role",this.form).then(res =>{  if(res.code ==='200'){  this.$message.success("保存成功")  this.dialogFormVisible=false  this.load()  }  else {  this.$message.error("保存失败")  }  })  },  saveRoleMenu(){  this.request.post("/role/roleMenu/" + this.roleId, this.$refs.tree.getCheckedKeys()).then(res =>{  if(res.code ==='200'){  this.$message.success("绑定成功")  this.menuDialogVis=false  }else {  this.$message.error(res.msg)  }  })  },  handleAdd(){  this.dialogFormVisible=true  this.form={}  },  handleEdit(row){  //this.form=row //赋予表单数据,这样有个bug就是，在点击取消的时候表单上仍然有值，不过数据库里面的值没有变化,  this.form=***JSON***.parse(***JSON***.stringify(row)); //这是解决办法  this.dialogFormVisible=true //打开对话框  },  del(id){  this.request.delete("/role/"+id).then(res =>{  if(res.code ==='200'){  this.$message.success("删除成功")  this.load()  }  else {  this.$message.error("删除失败")  }  })  },   delBatch(){  let ids = this.multipleSelection.map(v =>v.id) //[{},{},{}]=>[1,2,3]  this.request.post("/role/del/batch",ids).then(res =>{  if(res.code ==='200'){  this.$message.success("批量删除成功")  this.load()  }  else {  this.$message.error("批量删除失败")  }  })  },  reset(){  this.name=""  this.load()  },  handleSizeChange(pageSize){  ***console***.log(pageSize)  this.pageSize=pageSize  this.load()  },  handleCurrentChange(pageNum){  ***console***.log(pageNum)  this.pageNum=pageNum  this.load()  },  selectMenu(roleId){  this.menuDialogVis = true  this.roleId = roleId  //请求菜单数据  this.request.get("/menu").then(res =>{  this.menuData = res.data   //把后台返回的菜单数据处理成id数组  this.expends = this.menuData.map(v =>v.id)  })   this.request.get("/role/roleMenu/"+roleId).then(res =>{  this.checks = res.data   })  },  handleCheckChange(data, checked, indeterminate) {  ***console***.log(data, checked, indeterminate);  },  } } </script>  <style> .headerBg {  background: #eee**!important**; } </style> |

## Dict.java

|  |
| --- |
| package com.txx.springboot.entity;  import com.baomidou.mybatisplus.annotation.TableName; import lombok.Data;  @TableName("sys\_dict") @Data public class Dict {  private String name;  private String value;  private String type;  } |

## RoleMenu.java

|  |
| --- |
| package com.txx.springboot.entity;   import com.baomidou.mybatisplus.annotation.TableName; import lombok.Data;  @TableName("sys\_role\_menu") @Data public class RoleMenu {   private Integer roleId;  private Integer menuId; } |

## MenuController.java

|  |
| --- |
| package com.txx.springboot.controller;   import com.baomidou.mybatisplus.extension.plugins.pagination.Page; import com.txx.springboot.common.Constants; import com.txx.springboot.entity.Dict; import com.txx.springboot.mapper.DictMapper; import org.springframework.web.bind.annotation.\*; import javax.annotation.Resource; import java.util.List; import java.util.stream.Collectors;  import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper; import com.txx.springboot.common.Result;  import com.txx.springboot.service.IMenuService; import com.txx.springboot.entity.Menu;  import org.springframework.web.bind.annotation.RestController;  */\*\*  \* <p>  \* 前端控制器  \* </p>  \*  \** ***@author*** *txx  \** ***@since*** *2022-08-22  \*/* @RestController @RequestMapping("/menu")  public class MenuController {    @Resource  private IMenuService menuService;   @Resource  private DictMapper dictMapper;   // 新增或者更新  @PostMapping  public Result save(@RequestBody Menu menu) {  menuService.saveOrUpdate(menu);  return Result.*success*();  }   @DeleteMapping("/{id}")  public Result delete(@PathVariable Integer id) {  menuService.removeById(id);  return Result.*success*();  }   @PostMapping("/del/batch")  public Result deleteBatch(@RequestBody List<Integer> ids) {  menuService.removeByIds(ids);  return Result.*success*();  }   @GetMapping  public Result findAll(@RequestParam(defaultValue = "") String name) {  QueryWrapper<Menu> queryWrapper = new QueryWrapper<>();  queryWrapper.like("name",name);  //查询所有数据  List<Menu> list = menuService.list(queryWrapper);  //找出pid为null的一级菜单  List<Menu> parentNode = list.stream().filter(menu -> menu.getPid() == null).collect(Collectors.*toList*());  for(Menu menu : parentNode){  //筛选所有数据中pid=父级id的数据就是二级菜单  menu.setChildren(list.stream().filter(m -> menu.getId().equals(m.getPid())).collect(Collectors.*toList*()));  }  return Result.*success*(parentNode);  }   @GetMapping("/{id}")  public Result findOne(@PathVariable Integer id) {  return Result.*success*(menuService.getById(id));  }   @GetMapping("/page")  public Result findPage(@RequestParam String name,  @RequestParam Integer pageNum,  @RequestParam Integer pageSize) {  QueryWrapper<Menu> queryWrapper = new QueryWrapper<>();  queryWrapper.like("name",name);  queryWrapper.orderByDesc("id");  return Result.*success*(menuService.page(new Page<>(pageNum, pageSize), queryWrapper));  }    @GetMapping("/icons")  public Result getIcons(){  QueryWrapper<Dict> queryWrapper = new QueryWrapper<>();  queryWrapper.eq("type", Constants.*DICT\_TYPE\_ICON*);  return Result.*success*(dictMapper.selectList(null));  }   } |

## RoleController.java

|  |
| --- |
| package com.txx.springboot.controller;   import com.baomidou.mybatisplus.extension.plugins.pagination.Page; import org.springframework.web.bind.annotation.\*; import javax.annotation.Resource; import java.util.List; import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper; import com.txx.springboot.common.Result;  import com.txx.springboot.service.IRoleService; import com.txx.springboot.entity.Role;  import org.springframework.web.bind.annotation.RestController;  */\*\*  \* <p>  \* 前端控制器  \* </p>  \*  \** ***@author*** *txx  \** ***@since*** *2022-08-22  \*/* @RestController @RequestMapping("/role")  public class RoleController {    @Resource  private IRoleService roleService;   // 新增或者更新  @PostMapping  public Result save(@RequestBody Role role) {  roleService.saveOrUpdate(role);  return Result.*success*();  }   @DeleteMapping("/{id}")  public Result delete(@PathVariable Integer id) {  roleService.removeById(id);  return Result.*success*();  }   @PostMapping("/del/batch")  public Result deleteBatch(@RequestBody List<Integer> ids) {  roleService.removeByIds(ids);  return Result.*success*();  }   @GetMapping  public Result findAll() {  return Result.*success*(roleService.list());  }   @GetMapping("/{id}")  public Result findOne(@PathVariable Integer id) {  return Result.*success*(roleService.getById(id));  }   @GetMapping("/page")  public Result findPage(@RequestParam String name,  @RequestParam Integer pageNum,  @RequestParam Integer pageSize) {  QueryWrapper<Role> queryWrapper = new QueryWrapper<>();  queryWrapper.like("name",name);  queryWrapper.orderByDesc("id");  return Result.*success*(roleService.page(new Page<>(pageNum, pageSize), queryWrapper));  }   */\*\*  \* 绑定角色和菜单的关系  \** ***@param*** *roleId 角色id  \** ***@param*** *menuIds 菜单id数组  \** ***@return*** *\*/* @PostMapping("/roleMenu/{roleId}")  public Result roleMenu(@PathVariable Integer roleId,@RequestBody List<Integer> menuIds) {  roleService.setRoleMenu(roleId,menuIds);  return Result.*success*();  }  @GetMapping ("/roleMenu/{roleId}")  public Result getRoleMenu(@PathVariable Integer roleId) {  return Result.*success*(roleService.getRoleMenu(roleId));  } } |

## IRoleService.java

|  |
| --- |
| package com.txx.springboot.service;  import com.txx.springboot.entity.Role; import com.baomidou.mybatisplus.extension.service.IService;  import java.util.List;  */\*\*  \* <p>  \* 服务类  \* </p>  \*  \** ***@author*** *txx  \** ***@since*** *2022-08-22  \*/* public interface IRoleService extends IService<Role> {   void setRoleMenu(Integer roleId, List<Integer> menuIds);   List<Integer> getRoleMenu(Integer roleId); } |

## RoleServiceImpl.java

|  |
| --- |
| package com.txx.springboot.service.impl;  import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper; import com.txx.springboot.entity.Role; import com.txx.springboot.entity.RoleMenu; import com.txx.springboot.mapper.RoleMapper; import com.txx.springboot.mapper.RoleMenuMapper; import com.txx.springboot.service.IRoleService; import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl; import org.springframework.stereotype.Service; import org.springframework.transaction.annotation.Transactional;  import javax.annotation.Resource; import java.util.List;  */\*\*  \* <p>  \* 服务实现类  \* </p>  \*  \** ***@author*** *txx  \** ***@since*** *2022-08-22  \*/* @Service public class RoleServiceImpl extends ServiceImpl<RoleMapper, Role> implements IRoleService {   @Resource  private RoleMenuMapper roleMenuMapper;   @Transactional  @Override  public void setRoleMenu(Integer roleId, List<Integer> menuIds) { // QueryWrapper<RoleMenu> queryWrapper = new QueryWrapper<>(); // queryWrapper.eq("role\_id",roleId); // roleMenuMapper.delete(queryWrapper);    //先删后增  //先删除当前角色id所有的绑定关系  roleMenuMapper.deleteByRoleId(roleId);   //再把前端传过来的菜单id数组绑定到当前的这个角色id上去  for(Integer menuId : menuIds){  RoleMenu roleMenu = new RoleMenu();  roleMenu.setRoleId(roleId);  roleMenu.setMenuId(menuId);  roleMenuMapper.insert(roleMenu);  }  }   @Override  public List<Integer> getRoleMenu(Integer roleId) {  return roleMenuMapper.selectByRoleId(roleId);  } } |

## DictMapper.java

|  |
| --- |
| package com.txx.springboot.mapper;  import com.baomidou.mybatisplus.core.mapper.BaseMapper; import com.txx.springboot.entity.Dict;  public interface DictMapper extends BaseMapper<Dict> { } |

## RoleMenuMapper.java

|  |
| --- |
| package com.txx.springboot.mapper;  import com.baomidou.mybatisplus.core.mapper.BaseMapper; import com.txx.springboot.entity.RoleMenu; import org.apache.ibatis.annotations.Delete; import org.apache.ibatis.annotations.Param; import org.apache.ibatis.annotations.Select;  import java.util.List;  public interface RoleMenuMapper extends BaseMapper<RoleMenu> {   @Delete("delete from sys\_role\_menu where role\_id = #{roleId}")  int deleteByRoleId(@Param("roleId") Integer roleId);   @Select("select menu\_id from sys\_role\_menu where role\_id = #{roleId}")  List<Integer> selectByRoleId(@Param("roleId")Integer roleId); } |