package com.sdocean.dictionary.model;

public class PublicModel {

private int id;

private String parentCode;

private int classId;

private String classCode;

private String className;

private String value;

private String remark;

private String orderCode;

private String ids;

public String getIds() {

return ids;

}

public void setIds(String ids) {

this.ids = ids;

}

public String getOrderCode() {

return orderCode;

}

public void setOrderCode(String orderCode) {

this.orderCode = orderCode;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getParentCode() {

return parentCode;

}

public void setParentCode(String parentCode) {

this.parentCode = parentCode;

}

public int getClassId() {

return classId;

}

public void setClassId(int classId) {

this.classId = classId;

}

public String getClassCode() {

return classCode;

}

public void setClassCode(String classCode) {

this.classCode = classCode;

}

public String getClassName() {

return className;

}

public void setClassName(String className) {

this.className = className;

}

public String getValue() {

return value;

}

public void setValue(String value) {

this.value = value;

}

public String getRemark() {

return remark;

}

public void setRemark(String remark) {

this.remark = remark;

}

}

package com.sdocean.dictionary.model;

public class UnitGroupModel {

private int id;

private String code;

private String name;

private String description;

private int isactive;

private String isactiveName;

public String getIsactiveName() {

return isactiveName;

}

public void setIsactiveName(String isactiveName) {

this.isactiveName = isactiveName;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public int getIsactive() {

return isactive;

}

public void setIsactive(int isactive) {

this.isactive = isactive;

}

}

package com.sdocean.dictionary.model;

public class UnitModel {

private int id;

private String code;

private int pid;

private int groupId;

private String groupName;

private String name;

private String description;

private String logo;

private int isactive;

private String isactiveName;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public int getPid() {

return pid;

}

public void setPid(int pid) {

this.pid = pid;

}

public int getGroupId() {

return groupId;

}

public void setGroupId(int groupId) {

this.groupId = groupId;

}

public String getGroupName() {

return groupName;

}

public void setGroupName(String groupName) {

this.groupName = groupName;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public String getLogo() {

return logo;

}

public void setLogo(String logo) {

this.logo = logo;

}

public int getIsactive() {

return isactive;

}

public void setIsactive(int isactive) {

this.isactive = isactive;

}

public String getIsactiveName() {

return isactiveName;

}

public void setIsactiveName(String isactiveName) {

this.isactiveName = isactiveName;

}

}

package com.sdocean.dictionary.model;

public class UnitModel {

private int id;

private String code;

private int pid;

private int groupId;

private String groupName;

private String name;

private String description;

private String logo;

private int isactive;

private String isactiveName;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public int getPid() {

return pid;

}

public void setPid(int pid) {

this.pid = pid;

}

public int getGroupId() {

return groupId;

}

public void setGroupId(int groupId) {

this.groupId = groupId;

}

public String getGroupName() {

return groupName;

}

public void setGroupName(String groupName) {

this.groupName = groupName;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public String getLogo() {

return logo;

}

public void setLogo(String logo) {

this.logo = logo;

}

public int getIsactive() {

return isactive;

}

public void setIsactive(int isactive) {

this.isactive = isactive;

}

public String getIsactiveName() {

return isactiveName;

}

public void setIsactiveName(String isactiveName) {

this.isactiveName = isactiveName;

}

}

package com.sdocean.dictionary.action;

import java.util.List;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Logger;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.servlet.ModelAndView;

import com.sdocean.common.model.Result;

import com.sdocean.dictionary.model.PublicModel;

import com.sdocean.dictionary.service.PublicService;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.service.OperationLogService;

import com.sdocean.page.model.PageResult;

import com.sdocean.page.model.UiColumn;

@Controller

public class PublicAction {

private static Logger log = Logger.getLogger(PublicAction.class);

@Autowired

private PublicService publicService;

@Resource

OperationLogService logService;

@Autowired

private ConfigInfo info;

@RequestMapping("info\_public.do")

public ModelAndView info\_public(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/dictionary/publicInfo");

return mav;

}

/\*

\* 通过parentCode查询到公共代码列表

\*/

@RequestMapping(value="getPublicList.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getPublicList(@ModelAttribute("model") PublicModel model,HttpServletRequest request,

HttpServletResponse response){

List<PublicModel> publics = publicService.getPublicsByParent(model.getParentCode());

return JsonUtil.toJson(publics);

}

/\*

\* 为公共代码管理查询结果

\*/

@RequestMapping(value="showPublics.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String showPublics(@ModelAttribute("model") PublicModel model,HttpServletRequest request,

HttpServletResponse response){

PageResult result = new PageResult();

//为查询结果增加表头

List<UiColumn> cols = publicService.getCols4List();

result.setCols(cols);

List<PublicModel> rows = publicService.getPublics(model);

result.setRows(rows);

return JsonUtil.toJson(result);

}

/\*

\* 保存修改的公共代码

\*/

@RequestMapping(value="savePublicChange.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String savePublicChange(@ModelAttribute("model") PublicModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = publicService.savePublicChange(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

/\*

\* 保存新增的公共代码

\*/

@RequestMapping(value="saveNewPublic.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveNewPublic(@ModelAttribute("model") PublicModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = publicService.saveNewPublic(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

/\*

\* 删除选中的编码

\*/

@RequestMapping(value="delePublics.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String delePublics(@ModelAttribute("model") PublicModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = publicService.delePublic(model.getIds());

logService.saveOperationLog(result,request);

return result.getMessage();

}

}

package com.sdocean.dictionary.action;

import java.util.List;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.servlet.ModelAndView;

import com.sdocean.common.model.Result;

import com.sdocean.dictionary.model.UnitModel;

import com.sdocean.dictionary.service.UnitService;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.service.OperationLogService;

import com.sdocean.page.model.PageResult;

import com.sdocean.page.model.UiColumn;

@Controller

public class UnitAction {

@Resource

UnitService unitService;

@Resource

OperationLogService logService;

@Autowired

private ConfigInfo info;

@RequestMapping("info\_unit.do")

public ModelAndView info\_unit(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/dictionary/unitInfo");

return mav;

}

@RequestMapping(value="showUnitList.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String showUnitList(@ModelAttribute("model") UnitModel model,HttpServletRequest request,

HttpServletResponse response){

PageResult result = new PageResult();

//为查询结果增加表头

List<UiColumn> cols = unitService.getCols4GroupList();

result.setCols(cols);

//为查询结果增加列表

List<UnitModel> rows = unitService.getUnitList(model);

result.setRows(rows);

return JsonUtil.toJson(result);

}

/\*

\* 获得单位列表

\*/

@RequestMapping(value="getUnitList.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getUnitList(@ModelAttribute("model") UnitModel model,HttpServletRequest request,

HttpServletResponse response){

List<UnitModel> rows = unitService.getUnitList(model);

return JsonUtil.toJson(rows);

}

/\*

\* 保存修改信息

\*/

@RequestMapping(value="saveUnitChange.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveUnitChange(@ModelAttribute("model") UnitModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = unitService.saveUnitChange(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

/\*

\* 保存新增信息

\*/

@RequestMapping(value="saveNewUnit.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveNewUnit(@ModelAttribute("model") UnitModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = unitService.saveNewUnit(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

}

package com.sdocean.dictionary.action;

import java.util.ArrayList;

import java.util.List;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.servlet.ModelAndView;

import com.sdocean.common.model.Result;

import com.sdocean.dictionary.model.UnitGroupModel;

import com.sdocean.dictionary.service.UnitGroupService;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.service.OperationLogService;

import com.sdocean.page.model.PageResult;

import com.sdocean.page.model.UiColumn;

@Controller

public class UnitGroupAction {

@Resource

UnitGroupService unitGroupService;

@Resource

OperationLogService logService;

@Autowired

private ConfigInfo info;

@RequestMapping("info\_unit\_group.do")

public ModelAndView info\_unit\_group(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/dictionary/unitGroup");

return mav;

}

@RequestMapping(value="showGroupList.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String showGroupList(@ModelAttribute("model") UnitGroupModel model,HttpServletRequest request,

HttpServletResponse response){

PageResult result = new PageResult();

//为查询结果增加表头

List<UiColumn> cols = unitGroupService.getCols4GroupList();

result.setCols(cols);

//为查询结果增加列表

List<UnitGroupModel> rows = unitGroupService.getUnitGroups(model);

result.setRows(rows);

return JsonUtil.toJson(result);

}

/\*

\* 保存修改信息

\*/

@RequestMapping(value="saveGroupChange.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveGroupChange(@ModelAttribute("model") UnitGroupModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = unitGroupService.saveGroupChange(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

/\*

\* 保存新增信息

\*/

@RequestMapping(value="saveNewGroup.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveNewGroup(@ModelAttribute("model") UnitGroupModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = unitGroupService.saveNewGroup(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

/\*

\* 获得有效的单位组的集合

\*/

@RequestMapping(value="getUnitGroups.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getUnitGroups(HttpServletRequest request,

HttpServletResponse response){

List<UnitGroupModel> list = new ArrayList<UnitGroupModel>();

list = unitGroupService.getUnitGroups();

return JsonUtil.toJson(list);

}

}

package com.sdocean.dictionary.action;

import java.util.List;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.servlet.ModelAndView;

import com.sdocean.common.model.Result;

import com.sdocean.dictionary.model.WaterQualityStandardModel;

import com.sdocean.dictionary.service.WaterQualityStandardService;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.service.OperationLogService;

import com.sdocean.page.model.PageResult;

import com.sdocean.page.model.UiColumn;

@Controller

public class WaterQualityStandardAction {

@Resource

WaterQualityStandardService standardService;

@Resource

OperationLogService logService;

@Autowired

private ConfigInfo info;

@RequestMapping("info\_water\_level.do")

public ModelAndView info\_water\_level(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/dictionary/waterLevelInfo");

return mav;

}

@RequestMapping(value="getStandardList.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStandardList(@ModelAttribute("model") WaterQualityStandardModel model,HttpServletRequest request,

HttpServletResponse response){

PageResult result = new PageResult();

//为查询结果增加表头

List<UiColumn> cols = standardService.getCols4GroupList();

result.setCols(cols);

//为查询结果增加列表

List<WaterQualityStandardModel> rows = standardService.getStandardList(model);

result.setRows(rows);

return JsonUtil.toJson(result);

}

/\*

\* 保存修改信息

\*/

@RequestMapping(value="saveQualityChange.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveQualityChange(@ModelAttribute("model") WaterQualityStandardModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = standardService.saveQualityChange(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

/\*

\* 保存新增信息

\*/

@RequestMapping(value="saveNewQuality.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveNewQuality(@ModelAttribute("model") WaterQualityStandardModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = standardService.saveNewQuality(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

/\*

\* 删除选中的水质等级

\*/

@RequestMapping(value="deleWaterQulity.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String deleWaterQulity(@ModelAttribute("model") WaterQualityStandardModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = standardService.deleWaterQulity(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

}

package com.sdocean.device.service;

import java.util.ArrayList;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.SelectTree;

import com.sdocean.dataQuery.model.DataQueryModel;

import com.sdocean.device.dao.DeviceDao;

import com.sdocean.device.model.DeviceGroup;

import com.sdocean.device.model.DeviceModel;

import com.sdocean.page.model.UiColumn;

import com.sdocean.station.model.StationModel;

@Service

@Transactional(rollbackFor=Exception.class, propagation=Propagation.REQUIRED)

public class DeviceService {

@Autowired

private DeviceDao deviceDao;

/\*

\* 为公共代码管理提供表头

\*/

public List<UiColumn> getCols4List(){

List<UiColumn> cols = new ArrayList<UiColumn>();

UiColumn col1 = new UiColumn("id", "id", true, "\*");

UiColumn col2 = new UiColumn("设备名称", "name", true, "\*");

UiColumn col3 = new UiColumn("设备编码", "code", true, "\*");

UiColumn col4 = new UiColumn("设备模型", "deviceModel", true, "\*");

UiColumn col5 = new UiColumn("介绍", "brief", true, "\*");

UiColumn col6 = new UiColumn("明细", "detail", true, "\*");

UiColumn col19 = new UiColumn("orderCode", "orderCode", true, "\*");

UiColumn col10 = new UiColumn("保留位数", "pointNum", true, "\*");

UiColumn col12 = new UiColumn("维护周期(天)", "mainnum", true, "\*");

UiColumn col11 = new UiColumn("监测参数", "indicatorNames", true, "\*");

UiColumn col13 = new UiColumn("groupId", "groupId", false, "\*");

UiColumn col14 = new UiColumn("设备分类", "groupName", true, "\*");

UiColumn col15 = new UiColumn("typeId", "typeId", false, "\*");

UiColumn col16 = new UiColumn("设备类型", "typeName", true, "\*");

cols.add(col1);

cols.add(col2);

cols.add(col3);

cols.add(col6);

cols.add(col4);

cols.add(col5);

cols.add(col19);

cols.add(col10);

cols.add(col12);

cols.add(col11);

cols.add(col13);

cols.add(col14);

cols.add(col15);

cols.add(col16);

return cols;

}

/\*

\* 查询符合条件的设备列表

\*/

public List<DeviceModel> getDevices(DeviceModel model){

return deviceDao.getDevices(model);

}

/\*

\* 保存修改的设备信息

\*/

public Result saveDeviceChange(DeviceModel model){

return deviceDao.saveDeviceChange(model);

}

/\*

\* 保存新增的设备

\*/

public Result saveNewDevice(DeviceModel model){

return deviceDao.saveNewDevice(model);

}

/\*

\* 获得当前站点所关联的设备列表

\*/

public List<SelectTree> getDeviceListByStation(StationModel model){

return deviceDao.getDeviceListByStation(model);

}

/\*

\* 获得当前站点下的设备的列表

\*/

public List<DeviceModel> getDevices4Station(StationModel model){

return deviceDao.getDevices4Station(model);

}

/\*

\* 获得当前站点下有展示权限的设备的列表

\*/

public List<DeviceModel> getDevices4Station4Show(StationModel model){

return deviceDao.getDevices4Station4Show(model);

}

/\*

\* 为站点组列表提供表头

\*/

public List<UiColumn> getCols4GroupList(){

List<UiColumn> cols = new ArrayList<UiColumn>();

UiColumn col1 = new UiColumn("id", "id", true, "\*");

UiColumn col2 = new UiColumn("分类名称", "title", true, "\*");

UiColumn col3 = new UiColumn("分类编码", "code", true, "\*");

UiColumn col6 = new UiColumn("明细", "detail", true, "\*");

UiColumn col19 = new UiColumn("排序码", "ordercode", true, "\*");

cols.add(col1);

cols.add(col2);

cols.add(col3);

cols.add(col6);

cols.add(col19);

return cols;

}

/\*

\* 查询所有符合条件的设备组

\*/

public List<DeviceGroup> getDeviceGroupList(DeviceGroup model){

return deviceDao.getDeviceGroupList(model);

}

/\*

\* 新增设备组

\*/

public Result addDeviceGroup(DeviceGroup model){

return deviceDao.addDeviceGroup(model);

}

/\*

\* 修改设备组

\*/

public Result saveDeviceGroup(DeviceGroup model){

return deviceDao.saveDeviceGroup(model);

}

}

package com.sdocean.device.model;

public class DeviceGroup {

private int id;

private String code;

private String title;

private String detail;

private String ordercode;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getDetail() {

return detail;

}

public void setDetail(String detail) {

this.detail = detail;

}

public String getOrdercode() {

return ordercode;

}

public void setOrdercode(String ordercode) {

this.ordercode = ordercode;

}

}

package com.sdocean.device.model;

import java.util.List;

import com.sdocean.indicator.model.IndicatorModel;

public class DeviceModel {

private int id;

private String name;

private String code;

private String deviceModel;

private String brief;

private String detail;

private String orderCode;

private String indicatorIds;

private String indicatorNames;

private List<IndicatorModel> indicators;

private Integer interval;

private Integer pointNum; //小数点保留位数

private int mainnum; //例行维护周期

private int groupId;

private String groupName; //设备分组 device\_group表

private int typeId;

private String typeName; //设备类型 1是传感器 2是其他

public int getGroupId() {

return groupId;

}

public void setGroupId(int groupId) {

this.groupId = groupId;

}

public String getGroupName() {

return groupName;

}

public void setGroupName(String groupName) {

this.groupName = groupName;

}

public int getTypeId() {

return typeId;

}

public void setTypeId(int typeId) {

this.typeId = typeId;

}

public String getTypeName() {

return typeName;

}

public void setTypeName(String typeName) {

this.typeName = typeName;

}

public int getMainnum() {

return mainnum;

}

public void setMainnum(int mainnum) {

this.mainnum = mainnum;

}

public Integer getPointNum() {

return pointNum;

}

public void setPointNum(Integer pointNum) {

this.pointNum = pointNum;

}

public Integer getInterval() {

return interval;

}

public void setInterval(Integer interval) {

this.interval = interval;

}

public List<IndicatorModel> getIndicators() {

return indicators;

}

public void setIndicators(List<IndicatorModel> indicators) {

this.indicators = indicators;

}

public String getIndicatorIds() {

return indicatorIds;

}

public void setIndicatorIds(String indicatorIds) {

this.indicatorIds = indicatorIds;

}

public String getIndicatorNames() {

return indicatorNames;

}

public void setIndicatorNames(String indicatorNames) {

this.indicatorNames = indicatorNames;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getDeviceModel() {

return deviceModel;

}

public void setDeviceModel(String deviceModel) {

this.deviceModel = deviceModel;

}

public String getBrief() {

return brief;

}

public void setBrief(String brief) {

this.brief = brief;

}

public String getDetail() {

return detail;

}

public void setDetail(String detail) {

this.detail = detail;

}

public String getOrderCode() {

return orderCode;

}

public void setOrderCode(String orderCode) {

this.orderCode = orderCode;

}

}

package com.sdocean.device.dao;

import java.util.ArrayList;

import java.util.List;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.SelectTree;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.dataQuery.model.DataQueryModel;

import com.sdocean.device.model.DeviceGroup;

import com.sdocean.device.model.DeviceModel;

import com.sdocean.dictionary.model.PublicModel;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.indicator.model.IndicatorModel;

import com.sdocean.log.model.SysLoginLogModel;

import com.sdocean.menu.model.CurrMenu;

import com.sdocean.menu.model.SysMenu;

import com.sdocean.role.model.RoleModel;

import com.sdocean.station.model.StationModel;

import com.sdocean.users.model.SysUser;

@Component

public class DeviceDao extends OracleEngine{

/\*

\* 查询所有符合条件的设备组

\*/

public List<DeviceGroup> getDeviceGroupList(DeviceGroup model){

List<DeviceGroup> list = new ArrayList<DeviceGroup>();

StringBuffer sql = new StringBuffer("");

sql.append(" select id,code,title,detail,ordercode");

sql.append(" from device\_group ");

sql.append(" where 1=1");

//添加查询条件

if(model.getCode()!=null&&model.getCode().length()>0){

sql.append(" and (code like '%").append(model.getCode()).append("%' or title like '%").append(model.getCode()).append("%')");

}

//添加排序条件

sql.append(" order by ordercode");

list = this.queryObjectList(sql.toString(), DeviceGroup.class);

return list;

}

/\*

\* 新增设备组

\*/

public Result addDeviceGroup(DeviceGroup model){

//初始化返回结果

Result result = new Result();

result.setDotype(result.ADD);

result.setModel(JsonUtil.toJson(model));

result.setResult(result.SUCCESS);

result.setMessage("新增成功");

StringBuffer sql = new StringBuffer("");

sql.append(" insert into device\_group(code,title,detail,ordercode)");

sql.append(" values(?,?,?,?)");

Object[] params = new Object[]{

model.getCode(),model.getTitle(),model.getDetail(),model.getOrdercode()

};

try {

this.update(sql.toString(), params);

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("新增失败，请重试");

}

return result;

}

/\*

\* 修改设备组

\*/

public Result saveDeviceGroup(DeviceGroup model){

//初始化返回结果

Result result = new Result();

result.setDotype(result.UPDATE);

result.setModel(JsonUtil.toJson(model));

result.setResult(result.SUCCESS);

result.setMessage("修改成功");

StringBuffer sql = new StringBuffer("");

sql.append(" update device\_group set code=?,title=?,detail=?,ordercode=?");

sql.append(" where id = ?");

Object[] params = new Object[]{

model.getCode(),model.getTitle(),model.getDetail(),model.getOrdercode(),model.getId()

};

try {

this.update(sql.toString(), params);

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("修改失败，请重试");

}

return result;

}

/\*

\* 查询符合条件的设备列表

\*/

public List<DeviceModel> getDevices(DeviceModel model){

List<DeviceModel> list = new ArrayList<DeviceModel>();

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.name,a.code,a.deviceModel,a.brief,a.detail,a.ordercode,");

sql.append(" group\_concat(m.title SEPARATOR '|') as indicatorNames,");

sql.append(" group\_concat(m.indicatorid) as indicatorIds,pointnum,mainnum,");

sql.append(" a.groupid,g.title as groupname,a.typeid,case a.typeid when 1 then '传感器' else '其他' end as typename");

sql.append(" from device\_catalog a left join (select b.catalogId,b.indicatorid,c.title");

sql.append(" from device\_catalog\_indicator b,dm\_indicator c");

sql.append(" where b.indicatorid = c.id and c.isactive = 1) m");

sql.append(" on a.id = m.catalogid ,device\_group g");

sql.append(" where a.groupid = g.id");

//添加查询条件

if(model!=null&&model.getCode()!=null&&model.getCode().length()>0){

sql.append(" and ( a.name like '%").append(model.getCode()).append("%' or");

sql.append(" a.code like '%").append(model.getCode()).append("%') ");

}

if(model!=null&&model.getGroupId()>0){

sql.append(" and a.groupid = ").append(model.getGroupId());

}

sql.append(" group by a.id");

//添加排序

sql.append(" order by ordercode ");

list = this.queryObjectList(sql.toString(), DeviceModel.class);

return list;

}

/\*

\* 根据设备ID,查询设备信息

\*/

public DeviceModel getDeviceByDeviceId(int deviceId){

DeviceModel device = new DeviceModel();

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.name,a.code,a.deviceModel,a.brief,a.detail,a.ordercode,");

sql.append(" group\_concat(m.title SEPARATOR '|') as indicatorNames,");

sql.append(" group\_concat(m.indicatorid) as indicatorIds,pointnum,mainnum");

sql.append(" from device\_catalog a left join (select b.catalogId,b.indicatorid,c.title");

sql.append(" from device\_catalog\_indicator b,dm\_indicator c");

sql.append(" where b.indicatorid = c.id and c.isactive = 1) m");

sql.append(" on a.id = m.catalogid");

sql.append(" where a.id =").append(deviceId);

sql.append(" limit 1");

device = this.queryObject(sql.toString(), DeviceModel.class);

return device;

}

/\*

\* 保存修改的设备信息

\*/

public Result saveDeviceChange(DeviceModel model){

Result result = new Result();

result.setDotype(result.UPDATE);

result.setModel(JsonUtil.toJson(model));

result.setResult(result.SUCCESS);

result.setMessage("修改成功");

//验证code 是否唯一

StringBuffer checkSql = new StringBuffer("");

checkSql.append(" select count(1) from device\_catalog where code ='").append(model.getCode()).append("' and id <>").append(model.getId());

int cou = 0;

try {

cou = this.queryForInt(checkSql.toString(), null);

} catch (Exception e) {

e.printStackTrace();

result.setResult(result.FAILED);

result.setMessage("验证唯一性时失败");

return result;

}

if(cou>0){

result.setResult(result.FAILED);

result.setMessage("CODE重复,违反唯一性原则");

return result;

}

//开始保存数据

StringBuffer sql = new StringBuffer("");

sql.append(" update device\_catalog set name=?,code=?,devicemodel=?,brief=?,detail=?,ordercode=?,pointNum=?,mainnum=?,groupid=?,typeid=? where id =?");

Object[] params = new Object[]{

model.getName(),model.getCode(),model.getDeviceModel(),

model.getBrief(),model.getDetail(),model.getOrderCode(),model.getPointNum(),model.getMainnum(),model.getGroupId(),model.getTypeId(),model.getId()

};

int res = 0;

try {

res = this.update(sql.toString(), params);

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("保存失败");

return result;

}

this.saveDeviceIndicator(model);

return result;

}

/\*

\* 保存新增的设备

\*/

public Result saveNewDevice(DeviceModel model){

Result result = new Result();

result.setDotype(result.UPDATE);

result.setModel(JsonUtil.toJson(model));

result.setResult(result.SUCCESS);

result.setMessage("新增成功");

//验证code 是否唯一

StringBuffer checkSql = new StringBuffer("");

checkSql.append(" select count(1) from device\_catalog where code ='").append(model.getCode()).append("' ");

int cou = 0;

try {

cou = this.queryForInt(checkSql.toString(), null);

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("验证唯一性时失败");

return result;

}

if(cou>0){

result.setResult(result.FAILED);

result.setMessage("CODE重复,违反唯一性原则");

return result;

}

//开始保存数据

StringBuffer sql = new StringBuffer("");

sql.append(" insert into device\_catalog(name,code,devicemodel,brief,detail,ordercode,pointnum,mainnum,groupid,typeid) values(?,?,?,?,?,?,?,?,?,?)");

Object[] params = new Object[]{

model.getName(),model.getCode(),model.getDeviceModel(),

model.getBrief(),model.getDetail(),model.getOrderCode(),model.getPointNum(),model.getMainnum(),model.getGroupId(),model.getTypeId()

};

int res = 0;

try {

res = this.update(sql.toString(), params);

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("新增失败");

return result;

}

//获得保存的ID值\

String last = "SELECT LAST\_INSERT\_ID()";

int lastId = this.queryForInt(last, null);

model.setId(lastId);

this.saveDeviceIndicator(model);

return result;

}

public void saveDeviceIndicator(DeviceModel model){

//删除设备的参数

StringBuffer deleSql = new StringBuffer("");

deleSql.append(" delete from device\_catalog\_indicator where catalogid = ").append(model.getId());

this.update(deleSql.toString(), null);

String ids = model.getIndicatorIds();

String[] idStrings = ids.split(",");

StringBuffer val = new StringBuffer("(0,0)");

for(String id:idStrings){

val.append(",(").append(model.getId()).append(",").append(id).append(")");

}

StringBuffer insertSql = new StringBuffer("");

insertSql.append(" insert into device\_catalog\_indicator(catalogid,indicatorid) values").append(val);

insertSql.append(" ON DUPLICATE KEY UPDATE indicatorid = indicatorid ");

this.update(insertSql.toString(), null);

}

/\*

\* 获得当前站点所关联的设备列表

\*/

public List<SelectTree> getDeviceListByStation(StationModel model){

List<SelectTree> list = new ArrayList<SelectTree>();

//添加设备列表的首层

SelectTree first = new SelectTree();

first.setId("0");

first.setName("设备");

first.setIsExpanded(true);

first.setIsActive(true);

//获得所有有效的设备的列表

List<SelectTree> children = new ArrayList<SelectTree>();

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.name, case when b.aiot\_watch\_point\_id is null then 'false' else 'true' end as selected");

sql.append(" from device\_catalog a");

sql.append(" left join map\_awp\_device\_catalog b on a.id = b.device\_catalog\_id");

sql.append(" and b.aiot\_watch\_point\_id = ").append(model.getId());

children = this.queryObjectList(sql.toString(), SelectTree.class);

//将设备列表添加的首层的child中

first.setChildren(children);

//将首层添加到结果集中

list.add(first);

return list;

}

/\*

\* 获得当前站点下的设备的列表

\*/

public List<DeviceModel> getDevices4Station(StationModel model){

List<DeviceModel> list = new ArrayList<DeviceModel>();

StringBuffer sql = new StringBuffer("");

sql.append(" select distinct a.id,a.name,a.code,a.deviceModel,a.brief,a.detail,a.ordercode");

sql.append(" from device\_catalog a,map\_awp\_device\_catalog b");

sql.append(" where a.id = b.device\_catalog\_id");

//添加查询条件

if(model!=null&&model.getId()>0){

sql.append(" and b.aiot\_watch\_point\_id =").append(model.getId());

}

sql.append(" order by a.ordercode");

list = this.queryObjectList(sql.toString(), DeviceModel.class);

return list;

}

/\*

\* 获得当前站点下有展示权限的设备的列表

\*/

public List<DeviceModel> getDevices4Station4Show(StationModel model){

List<DeviceModel> list = new ArrayList<DeviceModel>();

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.name,a.code,a.deviceModel,a.brief,a.detail,a.ordercode");

sql.append(" from view\_show\_station\_device a");

sql.append(" where a.stationid =").append(model.getId());

sql.append(" order by a.ordercode");

list = this.queryObjectList(sql.toString(), DeviceModel.class);

return list;

}

/\*

\*根据deviceid以及indicators

\*/

public DeviceModel getDeviceByid(String deviceId,String indicatorids){

DeviceModel model = new DeviceModel();

//查询出设备

StringBuffer devicesql = new StringBuffer("");

devicesql.append(" select a.id,a.name,a.code,a.deviceModel,a.brief,a.detail,a.ordercode");

devicesql.append(" from device\_catalog a where a.id = ").append(deviceId);

model = this.queryObject(devicesql.toString(), DeviceModel.class);

//查询出参数列表

StringBuffer indicatorSql = new StringBuffer("");

indicatorSql.append(" select a.id,a.code,a.title,a.groupId,a.unitId,a.description,a.isactive");

indicatorSql.append(" from dm\_indicator a where a.id in (").append(indicatorids).append(")");

List<IndicatorModel> indicators = new ArrayList<IndicatorModel>();

indicators = this.queryObjectList(indicatorSql.toString(), IndicatorModel.class);

if(indicators!=null&&indicators.size()>0){

model.setIndicators(indicators);

}

return model;

}

/\*

\* 读取站点内设备读取数据的时间间隔

\*/

public DeviceModel getInterval4device(int stationid,DeviceModel device){

StringBuffer sql = new StringBuffer("");

sql.append("select max(betweentime) from aiot\_station\_device\_comm where stationid in (0, ").append(stationid).append(") and deviceid in (0").append(device.getId()).append(")");

Integer inter = this.queryForInt(sql.toString(), null);

if(inter==null){

inter = 0;

}

device.setInterval(inter);

return device;

}

}

package com.sdocean.device.action;

import java.util.ArrayList;

import java.util.List;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.apache.log4j.Logger;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.servlet.ModelAndView;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.SelectTree;

import com.sdocean.device.model.DeviceGroup;

import com.sdocean.device.model.DeviceModel;

import com.sdocean.device.service.DeviceService;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.service.OperationLogService;

import com.sdocean.page.model.PageResult;

import com.sdocean.page.model.UiColumn;

import com.sdocean.station.model.StationModel;

@Controller

public class DeviceAction {

private static Logger log = Logger.getLogger(DeviceAction.class);

@Autowired

private DeviceService deviceService;

@Resource

OperationLogService logService;

@Autowired

private ConfigInfo info;

/\*

\* 跳转到设备组页面

\*/

@RequestMapping("info\_device\_group.do")

public ModelAndView info\_device\_group(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/device/deviceGroupInfo");

return mav;

}

/\*

\* 查询设备分组列表

\*/

@RequestMapping(value="showDeviceGroupList.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String showDeviceGroupList(@ModelAttribute("model") DeviceGroup model,HttpServletRequest request,

HttpServletResponse response){

PageResult result = new PageResult();

//为查询结果增加表头

List<UiColumn> cols = deviceService.getCols4GroupList();

result.setCols(cols);

List<DeviceGroup> rows = deviceService.getDeviceGroupList(model);

result.setRows(rows);

return JsonUtil.toJson(result);

}

/\*

\* 获得设备分组列表

\*/

@RequestMapping(value="getDeviceGroupList.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getDeviceGroupList(@ModelAttribute("model") DeviceGroup model,HttpServletRequest request,

HttpServletResponse response){

List<DeviceGroup> rows = deviceService.getDeviceGroupList(model);

return JsonUtil.toJson(rows);

}

/\*

\* 保存新增的设备分组

\*/

@RequestMapping(value="addDeviceGroup.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String addDeviceGroup(@ModelAttribute("model") DeviceGroup model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = deviceService.addDeviceGroup(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

/\*

\* 保存修改的设备分组

\*/

@RequestMapping(value="saveDeviceGroup.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveDeviceGroup(@ModelAttribute("model") DeviceGroup model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = deviceService.saveDeviceGroup(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

@RequestMapping("info\_device.do")

public ModelAndView info\_public(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/device/deviceInfo");

return mav;

}

/\*

\* 跳转到设备监控页面

\*/

@RequestMapping("info\_deviceWatch.do")

public ModelAndView info\_deviceWatch(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getPageVision()+"/device/deviceWatch");

return mav;

}

/\*

\* 为设备管理查询结果

\*/

@RequestMapping(value="showDevices.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String showDevices(@ModelAttribute("model") DeviceModel model,HttpServletRequest request,

HttpServletResponse response){

PageResult result = new PageResult();

//为查询结果增加表头

List<UiColumn> cols = deviceService.getCols4List();

result.setCols(cols);

List<DeviceModel> rows = deviceService.getDevices(model);

result.setRows(rows);

return JsonUtil.toJson(result);

}

/\*

\* 保存修改的公共代码

\*/

@RequestMapping(value="saveDeviceChange.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveDeviceChange(@ModelAttribute("model") DeviceModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = deviceService.saveDeviceChange(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

/\*

\* 保存新增的公共代码

\*/

@RequestMapping(value="saveNewDevice.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveNewDevice(@ModelAttribute("model") DeviceModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = deviceService.saveNewDevice(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

/\*

\* 保存新增的公共代码

\*/

@RequestMapping(value="getDeviceListByStation.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getDeviceListByStation(@ModelAttribute("model") StationModel model,HttpServletRequest request,

HttpServletResponse response){

List<SelectTree> list = new ArrayList<SelectTree>();

list = deviceService.getDeviceListByStation(model);

return JsonUtil.toJson(list);

}

/\*

\* 获得当前站点下的所有的设备的列表

\*/

@RequestMapping(value="getDevices4Station.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getDevices4Station(@ModelAttribute("model") StationModel model,HttpServletRequest request,

HttpServletResponse response){

List<DeviceModel> list = new ArrayList<DeviceModel>();

list = deviceService.getDevices4Station(model);

return JsonUtil.toJson(list);

}

/\*

\* 获得当前站点下有展示权限的设备的列表

\*/

@RequestMapping(value="getDevices4Station4Show.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getDevices4Station4Show(@ModelAttribute("model") StationModel model,HttpServletRequest request,

HttpServletResponse response){

List<DeviceModel> list = new ArrayList<DeviceModel>();

list = deviceService.getDevices4Station4Show(model);

return JsonUtil.toJson(list);

}

/\*

\* 查询所有的设备列表

\*/

@RequestMapping(value="getDevices.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getDevices(HttpServletRequest request,

HttpServletResponse response){

DeviceModel model = new DeviceModel();

List<DeviceModel> rows = deviceService.getDevices(model);

return JsonUtil.toJson(rows);

}

}

package com.sdocean.dataQuery.service;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.common.model.Result;

import com.sdocean.dataQuery.dao.DataQueryDao;

import com.sdocean.dataQuery.model.AiotModel;

import com.sdocean.dataQuery.model.DataChangeModel;

import com.sdocean.dataQuery.model.DataImportModel;

import com.sdocean.dataQuery.model.DataQueryDataModel;

import com.sdocean.dataQuery.model.DataQueryModel;

import com.sdocean.firstpage.model.LastMetaData;

import com.sdocean.firstpage.model.MetaFirstPage;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.metadata.dao.MetadataDao;

import com.sdocean.metadata.dao.MetadataTableDao;

import com.sdocean.metadata.model.MetadataModel;

import com.sdocean.metadata.model.MetadataTable;

import com.sdocean.page.model.UiColumn;

import com.sdocean.station.model.StationModel;

@Service

@Transactional(rollbackFor=Exception.class, propagation=Propagation.REQUIRED)

public class DataQueryService {

@Autowired

private DataQueryDao dataQueryDao;

@Autowired

private MetadataDao metaDataDao;

@Autowired

private MetadataTableDao tableDao;

/\*

\* 为实时数据添加表头

\*/

public List<UiColumn> getCols4DataQuery(DataQueryModel model){

return dataQueryDao.getCols4DataQuery(model);

}

/\*

\* 为实时数据查询提供结果

\*/

public List<Map<String, Object>> getRows4DataQuery(DataQueryModel model){

return dataQueryDao.getRows4DataQuery(model);

}

/\*

\* 查询出当前站点信息的最后实时数据

\* 为首页展示

\*/

public List<MetaFirstPage> getData4FirstPage(StationModel station){

//dataQueryDao.queryData();

return dataQueryDao.getData4FirstPage(station);

}

/\*

\* 为数据修改查询结果提供表头

\*/

public List<UiColumn> getCols4DataChange(DataChangeModel model){

List<UiColumn> cols = new ArrayList<UiColumn>();

UiColumn col1 = new UiColumn("id", "id", false, "\*", false, "number");

UiColumn col2 = new UiColumn("时间", "collect\_time", true, "\*", true, "");

//col2.setCellFilter("data:\"yyyy-MM-dd HH:mm:ss.0\"");

UiColumn col3 = new UiColumn("数值", "data", true, "\*", true, "number");

cols.add(col1);

cols.add(col2);

cols.add(col3);

return cols;

}

/\*public static void main(String[] args) {

List<UiColumn> cols = new ArrayList<UiColumn>();

UiColumn col1 = new UiColumn("id", "id", false, "\*", false, "number");

UiColumn col2 = new UiColumn("时间", "collect\_time", true, "\*", true, "");

//col2.setCellFilter("data:\"yyyy-MM-dd HH:mm:ss.0\"");

UiColumn col3 = new UiColumn("数值", "data", true, "\*", true, "number");

cols.add(col1);

cols.add(col2);

cols.add(col3);

}\*/

/\*

\* 为数据修改的查询提供结果

\*/

public List<MetadataModel> getResult4DataChangeshow(DataChangeModel model){

return dataQueryDao.getResult4DataChangeshow(model);

}

/\*

\* 保存修改

\*/

public Result saveChangeData(DataChangeModel model){

return dataQueryDao.saveChangeData(model);

}

/\*

\* 保存数据导入

\*/

public Result saveImportData(DataImportModel model){

Result result = new Result();

result.setDotype(Result.ADD);

result.setModel(JsonUtil.toJson(model));

result.setResult(Result.FAILED);

result.setMessage("导入数据成功");

List<DataImportModel> list = new ArrayList<DataImportModel>();

try {

list = (List<DataImportModel>) JsonUtil.fromJsons(model.getImportString(), DataImportModel.class);

} catch (Exception e) {

result.setResult(Result.FAILED);

result.setMessage("数据错误");

return result;

}

if(list==null||list.size()<1){

result.setResult(Result.FAILED);

result.setMessage("数据错误");

return result;

}

//将其他形式的元数据转换成标准元数据

List<MetadataModel> metas = new ArrayList<MetadataModel>();

metas = metaDataDao.changeMetadata(list, 1, "csvImport");

//整合数据采集时间

for(MetadataModel meta:metas){

meta = metaDataDao.changeNewMetadata(meta);

StationModel station = new StationModel();

station.setId(meta.getWpId());

//将标准元数据存入到基础元数据表中

MetadataTable metaTable = tableDao.getOneTable(station, meta.getCollect\_time(), 1);

try {

metaDataDao.saveMetaData(metaTable.getTableName(), meta);

} catch (Exception e) {

// TODO: handle exception

result.setResult(Result.FAILED);

return result;

}

//将标准元数据存入到综合元数据表中

MetadataTable synTable = tableDao.getOneTable(station, meta.getCollect\_time(), 2);

try {

metaDataDao.saveSynData(synTable.getTableName(), meta);

} catch (Exception e) {

// TODO: handle exception

result.setResult(Result.FAILED);

result.setMessage("导入综合元数据表错误");

return result;

}

}

return result;

}

/\*

\* 获得某个设备的表头

\*/

public String getCols4DataQuery(int deviceid){

return dataQueryDao.getCols4DataQuery(deviceid);

}

/\*

\* 获得某个设备的内容

\*/

public List<Map<String, Object>> getRows4DataQuery(int deviceid){

return dataQueryDao.getRows4DataQuery(deviceid);

}

/\*

\* 获得某个设备的图片合集

\*/

public List<DataQueryDataModel> getImages4DataQuery(int deviceid){

return dataQueryDao.getImages4DataQuery(deviceid);

}

public List<AiotModel> getAiotDatas(int deviceid){

return dataQueryDao.getAiotDatas(deviceid);

}

}

package com.sdocean.dataQuery.service;

import java.util.ArrayList;

import java.util.List;

import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.common.model.Result;

import com.sdocean.dataQuery.dao.DataQueryDao;

import com.sdocean.dataQuery.model.AiotModel;

import com.sdocean.dataQuery.model.DataChangeModel;

import com.sdocean.dataQuery.model.DataImportModel;

import com.sdocean.dataQuery.model.DataQueryDataModel;

import com.sdocean.dataQuery.model.DataQueryModel;

import com.sdocean.firstpage.model.LastMetaData;

import com.sdocean.firstpage.model.MetaFirstPage;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.metadata.dao.MetadataDao;

import com.sdocean.metadata.dao.MetadataTableDao;

import com.sdocean.metadata.model.MetadataModel;

import com.sdocean.metadata.model.MetadataTable;

import com.sdocean.page.model.UiColumn;

import com.sdocean.station.model.StationModel;

@Service

@Transactional(rollbackFor=Exception.class, propagation=Propagation.REQUIRED)

public class DataQueryService {

@Autowired

private DataQueryDao dataQueryDao;

@Autowired

private MetadataDao metaDataDao;

@Autowired

private MetadataTableDao tableDao;

/\*

\* 为实时数据添加表头

\*/

public List<UiColumn> getCols4DataQuery(DataQueryModel model){

return dataQueryDao.getCols4DataQuery(model);

}

/\*

\* 为实时数据查询提供结果

\*/

public List<Map<String, Object>> getRows4DataQuery(DataQueryModel model){

return dataQueryDao.getRows4DataQuery(model);

}

/\*

\* 查询出当前站点信息的最后实时数据

\* 为首页展示

\*/

public List<MetaFirstPage> getData4FirstPage(StationModel station){

//dataQueryDao.queryData();

return dataQueryDao.getData4FirstPage(station);

}

/\*

\* 为数据修改查询结果提供表头

\*/

public List<UiColumn> getCols4DataChange(DataChangeModel model){

List<UiColumn> cols = new ArrayList<UiColumn>();

UiColumn col1 = new UiColumn("id", "id", false, "\*", false, "number");

UiColumn col2 = new UiColumn("时间", "collect\_time", true, "\*", true, "");

//col2.setCellFilter("data:\"yyyy-MM-dd HH:mm:ss.0\"");

UiColumn col3 = new UiColumn("数值", "data", true, "\*", true, "number");

cols.add(col1);

cols.add(col2);

cols.add(col3);

return cols;

}

/\*public static void main(String[] args) {

List<UiColumn> cols = new ArrayList<UiColumn>();

UiColumn col1 = new UiColumn("id", "id", false, "\*", false, "number");

UiColumn col2 = new UiColumn("时间", "collect\_time", true, "\*", true, "");

//col2.setCellFilter("data:\"yyyy-MM-dd HH:mm:ss.0\"");

UiColumn col3 = new UiColumn("数值", "data", true, "\*", true, "number");

cols.add(col1);

cols.add(col2);

cols.add(col3);

}\*/

/\*

\* 为数据修改的查询提供结果

\*/

public List<MetadataModel> getResult4DataChangeshow(DataChangeModel model){

return dataQueryDao.getResult4DataChangeshow(model);

}

/\*

\* 保存修改

\*/

public Result saveChangeData(DataChangeModel model){

return dataQueryDao.saveChangeData(model);

}

/\*

\* 保存数据导入

\*/

public Result saveImportData(DataImportModel model){

Result result = new Result();

result.setDotype(Result.ADD);

result.setModel(JsonUtil.toJson(model));

result.setResult(Result.FAILED);

result.setMessage("导入数据成功");

List<DataImportModel> list = new ArrayList<DataImportModel>();

try {

list = (List<DataImportModel>) JsonUtil.fromJsons(model.getImportString(), DataImportModel.class);

} catch (Exception e) {

result.setResult(Result.FAILED);

result.setMessage("数据错误");

return result;

}

if(list==null||list.size()<1){

result.setResult(Result.FAILED);

result.setMessage("数据错误");

return result;

}

//将其他形式的元数据转换成标准元数据

List<MetadataModel> metas = new ArrayList<MetadataModel>();

metas = metaDataDao.changeMetadata(list, 1, "csvImport");

//整合数据采集时间

for(MetadataModel meta:metas){

meta = metaDataDao.changeNewMetadata(meta);

StationModel station = new StationModel();

station.setId(meta.getWpId());

//将标准元数据存入到基础元数据表中

MetadataTable metaTable = tableDao.getOneTable(station, meta.getCollect\_time(), 1);

try {

metaDataDao.saveMetaData(metaTable.getTableName(), meta);

} catch (Exception e) {

// TODO: handle exception

result.setResult(Result.FAILED);

return result;

}

//将标准元数据存入到综合元数据表中

MetadataTable synTable = tableDao.getOneTable(station, meta.getCollect\_time(), 2);

try {

metaDataDao.saveSynData(synTable.getTableName(), meta);

} catch (Exception e) {

// TODO: handle exception

result.setResult(Result.FAILED);

result.setMessage("导入综合元数据表错误");

return result;

}

}

return result;

}

/\*

\* 获得某个设备的表头

\*/

public String getCols4DataQuery(int deviceid){

return dataQueryDao.getCols4DataQuery(deviceid);

}

/\*

\* 获得某个设备的内容

\*/

public List<Map<String, Object>> getRows4DataQuery(int deviceid){

return dataQueryDao.getRows4DataQuery(deviceid);

}

/\*

\* 获得某个设备的图片合集

\*/

public List<DataQueryDataModel> getImages4DataQuery(int deviceid){

return dataQueryDao.getImages4DataQuery(deviceid);

}

public List<AiotModel> getAiotDatas(int deviceid){

return dataQueryDao.getAiotDatas(deviceid);

}

}

package com.sdocean.dataQuery.service;

import java.util.List;

import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.common.model.Result;

import com.sdocean.dataQuery.dao.DataQueryDao;

import com.sdocean.dataQuery.dao.MouldDao;

import com.sdocean.dataQuery.model.DataQueryModel;

import com.sdocean.dataQuery.model.MouldModel;

import com.sdocean.firstpage.model.LastMetaData;

import com.sdocean.page.model.NgColumn;

import com.sdocean.station.model.StationModel;

@Service

@Transactional(rollbackFor=Exception.class, propagation=Propagation.REQUIRED)

public class MouldService {

@Autowired

private MouldDao mouldDao;

/\*

\* 获得当前站点的水质评价模板

\*/

public MouldModel getMouldByStationId(MouldModel model){

return mouldDao.getMouldByStationId(model);

}

/\*

\* 保存当前站点的水质评价模板

\*/

public Result saveMouldSetting(MouldModel model){

return mouldDao.saveMouldSetting(model);

}

/\*

\* 根据站点,获得水质评价模板

\*/

public MouldModel getMouldByStation(StationModel model){

return mouldDao.getMouldByStation(model);

}

}

package com.sdocean.dataQuery.service;

import java.util.List;

import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.common.model.Result;

import com.sdocean.dataQuery.dao.DataQueryDao;

import com.sdocean.dataQuery.dao.MouldDao;

import com.sdocean.dataQuery.model.DataQueryModel;

import com.sdocean.dataQuery.model.MouldModel;

import com.sdocean.firstpage.model.LastMetaData;

import com.sdocean.page.model.NgColumn;

import com.sdocean.station.model.StationModel;

@Service

@Transactional(rollbackFor=Exception.class, propagation=Propagation.REQUIRED)

public class MouldService {

@Autowired

private MouldDao mouldDao;

/\*

\* 获得当前站点的水质评价模板

\*/

public MouldModel getMouldByStationId(MouldModel model){

return mouldDao.getMouldByStationId(model);

}

/\*

\* 保存当前站点的水质评价模板

\*/

public Result saveMouldSetting(MouldModel model){

return mouldDao.saveMouldSetting(model);

}

/\*

\* 根据站点,获得水质评价模板

\*/

public MouldModel getMouldByStation(StationModel model){

return mouldDao.getMouldByStation(model);

}

}

package com.sdocean.dataQuery.service;

import java.util.List;

import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.common.model.Result;

import com.sdocean.dataQuery.dao.DataQueryDao;

import com.sdocean.dataQuery.dao.MouldDao;

import com.sdocean.dataQuery.model.DataQueryModel;

import com.sdocean.dataQuery.model.MouldModel;

import com.sdocean.firstpage.model.LastMetaData;

import com.sdocean.page.model.NgColumn;

import com.sdocean.station.model.StationModel;

@Service

@Transactional(rollbackFor=Exception.class, propagation=Propagation.REQUIRED)

public class MouldService {

@Autowired

private MouldDao mouldDao;

/\*

\* 获得当前站点的水质评价模板

\*/

public MouldModel getMouldByStationId(MouldModel model){

return mouldDao.getMouldByStationId(model);

}

/\*

\* 保存当前站点的水质评价模板

\*/

public Result saveMouldSetting(MouldModel model){

return mouldDao.saveMouldSetting(model);

}

/\*

\* 根据站点,获得水质评价模板

\*/

public MouldModel getMouldByStation(StationModel model){

return mouldDao.getMouldByStation(model);

}

}

package com.sdocean.dataQuery.service;

import java.util.List;

import java.util.Map;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.common.model.Result;

import com.sdocean.dataQuery.dao.DataQueryDao;

import com.sdocean.dataQuery.dao.MouldDao;

import com.sdocean.dataQuery.model.DataQueryModel;

import com.sdocean.dataQuery.model.MouldModel;

import com.sdocean.firstpage.model.LastMetaData;

import com.sdocean.page.model.NgColumn;

import com.sdocean.station.model.StationModel;

@Service

@Transactional(rollbackFor=Exception.class, propagation=Propagation.REQUIRED)

public class MouldService {

@Autowired

private MouldDao mouldDao;

/\*

\* 获得当前站点的水质评价模板

\*/

public MouldModel getMouldByStationId(MouldModel model){

return mouldDao.getMouldByStationId(model);

}

/\*

\* 保存当前站点的水质评价模板

\*/

public Result saveMouldSetting(MouldModel model){

return mouldDao.saveMouldSetting(model);

}

/\*

\* 根据站点,获得水质评价模板

\*/

public MouldModel getMouldByStation(StationModel model){

return mouldDao.getMouldByStation(model);

}

}

package com.sdocean.dataQuery.model;

public class AiotModel {

private int id;

private int deviceid;

private String indicatorcode;

private String collecttime;

private String data;

private String unit;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getDeviceid() {

return deviceid;

}

public void setDeviceid(int deviceid) {

this.deviceid = deviceid;

}

public String getIndicatorcode() {

return indicatorcode;

}

public void setIndicatorcode(String indicatorcode) {

this.indicatorcode = indicatorcode;

}

public String getCollecttime() {

return collecttime;

}

public void setCollecttime(String collecttime) {

this.collecttime = collecttime;

}

public String getData() {

return data;

}

public void setData(String data) {

this.data = data;

}

public String getUnit() {

return unit;

}

public void setUnit(String unit) {

this.unit = unit;

}

}

package com.sdocean.dataQuery.model;

import com.sdocean.device.model.DeviceModel;

import com.sdocean.indicator.model.IndicatorModel;

import com.sdocean.page.model.PageResult;

import com.sdocean.station.model.StationModel;

public class DataChangeModel {

private int id;

private int stationId;

private StationModel station;

private int deviceId;

private DeviceModel device;

private String indicatorCode;

private IndicatorModel indicator;

private String indicatorIds;

private PageResult page;

private String collect\_time;

private double newData;

private double oldData;

private String beginDate;

private String endDate;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public StationModel getStation() {

return station;

}

public void setStation(StationModel station) {

this.station = station;

}

public DeviceModel getDevice() {

return device;

}

public void setDevice(DeviceModel device) {

this.device = device;

}

public IndicatorModel getIndicator() {

return indicator;

}

public void setIndicator(IndicatorModel indicator) {

this.indicator = indicator;

}

public String getBeginDate() {

return beginDate;

}

public void setBeginDate(String beginDate) {

this.beginDate = beginDate;

}

public String getEndDate() {

return endDate;

}

public void setEndDate(String endDate) {

this.endDate = endDate;

}

public PageResult getPage() {

return page;

}

public void setPage(PageResult page) {

this.page = page;

}

public String getIndicatorIds() {

return indicatorIds;

}

public void setIndicatorIds(String indicatorIds) {

this.indicatorIds = indicatorIds;

}

public int getStationId() {

return stationId;

}

public void setStationId(int stationId) {

this.stationId = stationId;

}

public int getDeviceId() {

return deviceId;

}

public void setDeviceId(int deviceId) {

this.deviceId = deviceId;

}

public String getIndicatorCode() {

return indicatorCode;

}

public void setIndicatorCode(String indicatorCode) {

this.indicatorCode = indicatorCode;

}

public String getCollect\_time() {

return collect\_time;

}

public void setCollect\_time(String collect\_time) {

this.collect\_time = collect\_time;

}

public double getNewData() {

return newData;

}

public void setNewData(double newData) {

this.newData = newData;

}

public double getOldData() {

return oldData;

}

public void setOldData(double oldData) {

this.oldData = oldData;

}

}

package com.sdocean.dataQuery.model;

public class DataImportModel {

private String importString;

private int stationId;

private int deviceId;

private String indicatorCode;

private String collect\_time;

private Double data;

public String getCollect\_time() {

return collect\_time;

}

public void setCollect\_time(String collect\_time) {

this.collect\_time = collect\_time;

}

public Double getData() {

return data;

}

public void setData(Double data) {

this.data = data;

}

public String getImportString() {

return importString;

}

public void setImportString(String importString) {

this.importString = importString;

}

public int getStationId() {

return stationId;

}

public void setStationId(int stationId) {

this.stationId = stationId;

}

public int getDeviceId() {

return deviceId;

}

public void setDeviceId(int deviceId) {

this.deviceId = deviceId;

}

public String getIndicatorCode() {

return indicatorCode;

}

public void setIndicatorCode(String indicatorCode) {

this.indicatorCode = indicatorCode;

}

}

package com.sdocean.dataQuery.model;

import java.util.List;

import java.util.Map;

import com.sdocean.page.model.UiColumn;

public class DataQueryData {

private List<UiColumn> cols;

private List<Map<String, Object>> rows;

private List<DataQueryDataModel> images;

private List<AiotModel> aiots;

public List<AiotModel> getAiots() {

return aiots;

}

public void setAiots(List<AiotModel> aiots) {

this.aiots = aiots;

}

public List<UiColumn> getCols() {

return cols;

}

public void setCols(List<UiColumn> cols) {

this.cols = cols;

}

public List<Map<String, Object>> getRows() {

return rows;

}

public void setRows(List<Map<String, Object>> rows) {

this.rows = rows;

}

public List<DataQueryDataModel> getImages() {

return images;

}

public void setImages(List<DataQueryDataModel> images) {

this.images = images;

}

}

package com.sdocean.dataQuery.model;

import java.util.List;

import java.util.Map;

import com.sdocean.page.model.UiColumn;

public class DataQueryData {

private List<UiColumn> cols;

private List<Map<String, Object>> rows;

private List<DataQueryDataModel> images;

private List<AiotModel> aiots;

public List<AiotModel> getAiots() {

return aiots;

}

public void setAiots(List<AiotModel> aiots) {

this.aiots = aiots;

}

public List<UiColumn> getCols() {

return cols;

}

public void setCols(List<UiColumn> cols) {

this.cols = cols;

}

public List<Map<String, Object>> getRows() {

return rows;

}

public void setRows(List<Map<String, Object>> rows) {

this.rows = rows;

}

public List<DataQueryDataModel> getImages() {

return images;

}

public void setImages(List<DataQueryDataModel> images) {

this.images = images;

}

}

package com.sdocean.dataQuery.model;

import java.util.List;

import java.util.Map;

import com.sdocean.page.model.UiColumn;

public class DataQueryData {

private List<UiColumn> cols;

private List<Map<String, Object>> rows;

private List<DataQueryDataModel> images;

private List<AiotModel> aiots;

public List<AiotModel> getAiots() {

return aiots;

}

public void setAiots(List<AiotModel> aiots) {

this.aiots = aiots;

}

public List<UiColumn> getCols() {

return cols;

}

public void setCols(List<UiColumn> cols) {

this.cols = cols;

}

public List<Map<String, Object>> getRows() {

return rows;

}

public void setRows(List<Map<String, Object>> rows) {

this.rows = rows;

}

public List<DataQueryDataModel> getImages() {

return images;

}

public void setImages(List<DataQueryDataModel> images) {

this.images = images;

}

}

package com.sdocean.sms.model;

public class SmsSettingModel {

private int stationId;

private String stationName;

private int userId;

private String userIds;

private String userName;

private String telephone;

private int mon;

private String monValue;

private int tues;

private String tuesValue;

private int wed;

private String wedValue;

private int thur;

private String thurValue;

private int fri;

private String friValue;

private int satur;

private String saturValue;

private int sun;

private String sunValue;

private int betweenTime;

private int isactive;

private String isactiveName; //是否接收短信 1接收,0 不接受

private String ambegin;

private String amend;

private String pmbegin;

private String pmend;

private String type;

public String getUserIds() {

return userIds;

}

public void setUserIds(String userIds) {

this.userIds = userIds;

}

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

public int getIsactive() {

return isactive;

}

public void setIsactive(int isactive) {

this.isactive = isactive;

}

public String getIsactiveName() {

return isactiveName;

}

public void setIsactiveName(String isactiveName) {

this.isactiveName = isactiveName;

}

public String getPmbegin() {

return pmbegin;

}

public void setPmbegin(String pmbegin) {

this.pmbegin = pmbegin;

}

public String getAmbegin() {

return ambegin;

}

public void setAmbegin(String ambegin) {

this.ambegin = ambegin;

}

public String getAmend() {

return amend;

}

public void setAmend(String amend) {

this.amend = amend;

}

public String getPmend() {

return pmend;

}

public void setPmend(String pmend) {

this.pmend = pmend;

}

public String getTelephone() {

return telephone;

}

public void setTelephone(String telephone) {

this.telephone = telephone;

}

public int getStationId() {

return stationId;

}

public void setStationId(int stationId) {

this.stationId = stationId;

}

public String getStationName() {

return stationName;

}

public void setStationName(String stationName) {

this.stationName = stationName;

}

public int getUserId() {

return userId;

}

public void setUserId(int userId) {

this.userId = userId;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public int getMon() {

return mon;

}

public void setMon(int mon) {

this.mon = mon;

}

public String getMonValue() {

return monValue;

}

public void setMonValue(String monValue) {

this.monValue = monValue;

}

public int getTues() {

return tues;

}

public void setTues(int tues) {

this.tues = tues;

}

public String getTuesValue() {

return tuesValue;

}

public void setTuesValue(String tuesValue) {

this.tuesValue = tuesValue;

}

public int getWed() {

return wed;

}

public void setWed(int wed) {

this.wed = wed;

}

public String getWedValue() {

return wedValue;

}

public void setWedValue(String wedValue) {

this.wedValue = wedValue;

}

public int getThur() {

return thur;

}

public void setThur(int thur) {

this.thur = thur;

}

public String getThurValue() {

return thurValue;

}

public void setThurValue(String thurValue) {

this.thurValue = thurValue;

}

public int getFri() {

return fri;

}

public void setFri(int fri) {

this.fri = fri;

}

public String getFriValue() {

return friValue;

}

public void setFriValue(String friValue) {

this.friValue = friValue;

}

public int getSatur() {

return satur;

}

public void setSatur(int satur) {

this.satur = satur;

}

public String getSaturValue() {

return saturValue;

}

public void setSaturValue(String saturValue) {

this.saturValue = saturValue;

}

public int getSun() {

return sun;

}

public void setSun(int sun) {

this.sun = sun;

}

public String getSunValue() {

return sunValue;

}

public void setSunValue(String sunValue) {

this.sunValue = sunValue;

}

public int getBetweenTime() {

return betweenTime;

}

public void setBetweenTime(int betweenTime) {

this.betweenTime = betweenTime;

}

}

package com.sdocean.sms.service;

import java.util.ArrayList;

import java.util.List;

import javax.annotation.Resource;

import org.springframework.stereotype.Service;

import org.springframework.transaction.annotation.Propagation;

import org.springframework.transaction.annotation.Transactional;

import com.sdocean.common.model.Result;

import com.sdocean.page.model.UiColumn;

import com.sdocean.sms.dao.SmsDao;

import com.sdocean.sms.model.SmsLog;

import com.sdocean.sms.model.SmsMouldModel;

import com.sdocean.sms.model.SmsSettingModel;

import com.sdocean.users.model.SysUser;

@Service

@Transactional(rollbackFor = Exception.class, propagation = Propagation.REQUIRED)

public class SmsService {

@Resource

private SmsDao smsDao;

/\*

\* 为人员管理的查询结果添加表头

\*/

public List<UiColumn> getCols4TypeList(){

List<UiColumn> cols = new ArrayList<UiColumn>();

//UiColumn col1 = new UiColumn("stationId", "stationId", false, "\*");

//UiColumn col2 = new UiColumn("stationName", "stationName", true, "\*");

UiColumn col3 = new UiColumn("userId", "userId", false, "\*");

UiColumn col6 = new UiColumn("用户", "userName", true, "\*");

UiColumn col7 = new UiColumn("手机号", "telephone", true, "\*");

UiColumn col8 = new UiColumn("周一", "monValue", false, "\*");

UiColumn col9 = new UiColumn("周二", "tuesValue", false, "\*");

UiColumn col10 = new UiColumn("周三", "wedValue", false, "\*");

UiColumn col11 = new UiColumn("周四", "thurValue", false, "\*");

UiColumn col12 = new UiColumn("周五", "friValue", false, "\*");

UiColumn col13 = new UiColumn("周六", "saturValue", false, "\*");

UiColumn col14 = new UiColumn("周日", "sunValue", false, "\*");

UiColumn col15 = new UiColumn("周一", "mon", true, "\*");

UiColumn col16 = new UiColumn("周二", "tues", true, "\*");

UiColumn col17 = new UiColumn("周三", "wed", true, "\*");

UiColumn col21 = new UiColumn("周四", "thur", true, "\*");

UiColumn col18 = new UiColumn("周五", "fri", true, "\*");

UiColumn col19 = new UiColumn("周六", "satur", true, "\*");

UiColumn col20= new UiColumn("周日", "sun", true, "\*");

UiColumn col22= new UiColumn("间隔发送时间", "betweenTime", true, "\*");

UiColumn col27= new UiColumn("是否接收短信", "isactiveName", true, "\*");

UiColumn col23= new UiColumn("上午起始时间", "ambegin", true, "\*");

UiColumn col24= new UiColumn("上午截止时间", "amend", true, "\*");

UiColumn col25= new UiColumn("下午起始时间", "pmbegin", true, "\*");

UiColumn col26= new UiColumn("下午截止时间", "pmend", true, "\*");

cols.add(col3);

cols.add(col6);

cols.add(col7);

cols.add(col8);

cols.add(col9);

cols.add(col10);

cols.add(col11);

cols.add(col12);

cols.add(col13);

cols.add(col14);

cols.add(col15);

cols.add(col16);

cols.add(col17);

cols.add(col18);

cols.add(col19);

cols.add(col20);

cols.add(col21);

cols.add(col22);

cols.add(col23);

cols.add(col24);

cols.add(col25);

cols.add(col26);

cols.add(col27);

return cols;

}

/\*

\* 查询某站点下需要接受短信的人员列表

\*/

public List<SmsSettingModel> getUsers4SmsSetting(SmsSettingModel model,SysUser user){

return smsDao.getUsers4SmsSetting(model, user);

}

/\*

\* 初始化用户下的短信配置

\*/

public SmsSettingModel infoUserSetting(SysUser user){

return smsDao.infoUserSetting(user);

}

/\*

\* 保存用户短信配置

\*/

public Result saveUserSetting(SmsSettingModel model){

return smsDao.saveUserSetting(model);

}

/\*

\* 展示短信模板

\*/

public List<UiColumn> getCols4MouldList(){

List<UiColumn> cols = new ArrayList<UiColumn>();

UiColumn col1 = new UiColumn("id", "id", false, "\*");

UiColumn col2 = new UiColumn("stationId", "stationId", false, "\*");

UiColumn col3 = new UiColumn("站点", "stationName", true, "\*");

UiColumn col6 = new UiColumn("type", "type", false, "\*");

UiColumn col7 = new UiColumn("模板类型", "typeName", true, "\*");

UiColumn col8 = new UiColumn("模板", "mould", true, "\*");

cols.add(col1);

cols.add(col2);

cols.add(col3);

cols.add(col6);

cols.add(col7);

cols.add(col8);

return cols;

}

/\*

\* 为展示模板提供结果

\*/

public List<SmsMouldModel> getSmsMoulds4Rows(SmsMouldModel model){

return smsDao.getSmsMoulds4Rows(model);

}

/\*

\* 对短信发送模板修改

\*/

public Result saveSmsMouldChange(SmsMouldModel model){

return smsDao.saveSmsMouldChange(model);

}

/\*

\* 保存新增的短信发送模板

\*/

public Result saveNewSmsMould(SmsMouldModel model){

return smsDao.saveNewSmsMould(model);

}

/\*

\* 展示短信发送记录

\*/

public List<UiColumn> getCols4LogList(){

List<UiColumn> cols = new ArrayList<UiColumn>();

UiColumn col1 = new UiColumn("id", "id", false, "\*");

UiColumn col2 = new UiColumn("userId", "userId", false, "\*");

UiColumn col3 = new UiColumn("用户名", "userName", true, "\*");

UiColumn col6 = new UiColumn("手机号", "telephone", true, "\*");

UiColumn col7 = new UiColumn("短信", "message", true, "\*");

UiColumn col8 = new UiColumn("ifsend", "ifsend", false, "\*");

UiColumn col9 = new UiColumn("发送状态", "ifSendName", true, "\*");

cols.add(col1);

cols.add(col2);

cols.add(col3);

cols.add(col6);

cols.add(col7);

cols.add(col8);

cols.add(col9);

return cols;

}

/\*

\* 查询短信发送记录,

\* 增加模糊查询

\*/

public List<SmsLog> getSmsLogs(SmsLog model){

return smsDao.getSmsLogs(model);

}

/\*

\* 保存短信发送配置

\*/

public Result saveSmsStationSetting(SmsSettingModel model){

return smsDao.saveSmsStationSetting(model);

}

}

package com.sdocean.station.action;

import java.util.ArrayList;

import java.util.List;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.apache.log4j.Logger;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.servlet.ModelAndView;

import com.google.gson.JsonIOException;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.common.service.CommonService;

import com.sdocean.common.service.ZTreeService;

import com.sdocean.company.model.CompanyModel;

import com.sdocean.company.service.CompanyService;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.service.OperationLogService;

import com.sdocean.login.action.LoginAction;

import com.sdocean.page.model.PageResult;

import com.sdocean.page.model.UiColumn;

import com.sdocean.region.model.RegionModel;

import com.sdocean.region.service.RegionService;

import com.sdocean.role.model.RoleModel;

import com.sdocean.station.model.PlatformModel;

import com.sdocean.station.model.StationModel;

import com.sdocean.station.service.PlatformService;

import com.sdocean.station.service.StationService;

import com.sdocean.users.model.SysUser;

@Controller

public class PlatformAction {

private static Logger log = Logger.getLogger(PlatformAction.class);

@Resource

PlatformService platformService;

@Resource

OperationLogService logService;

@Autowired

private ConfigInfo info;

/\*

\* 跳转到平台编辑页面

\*/

@RequestMapping("info\_platform.do")

public ModelAndView info\_platform(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/platform/platformInfo");

return mav;

}

/\*

\* 展示平台列表

\*/

@RequestMapping(value="showPlatformList.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String showPlatformList(@ModelAttribute("model") PlatformModel model,HttpServletRequest request,

HttpServletResponse response){

PageResult result = new PageResult();

//为查询结果增加表头

List<UiColumn> cols = platformService.getCols4PlatformList();

result.setCols(cols);

//为查询结果增加列表

List<PlatformModel> rows = platformService.getPlatformList(model);

result.setRows(rows);

return JsonUtil.toJson(result);

}

/\*

\* 获得平台列表

\*/

@RequestMapping(value="getPlatformList.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getPlatformList(@ModelAttribute("model") PlatformModel model,HttpServletRequest request,

HttpServletResponse response){

List<PlatformModel> rows = platformService.getPlatformList(model);

return JsonUtil.toJson(rows);

}

/\*

\* 新增平台

\*/

@RequestMapping(value="addPlatform.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String addPlatform(@ModelAttribute("model") PlatformModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = platformService.addPlatform(model);

return result.getMessage();

}

/\*

\* 修改平台

\*/

@RequestMapping(value="savePlatform.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String savePlatform(@ModelAttribute("model") PlatformModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = platformService.savePlatform(model);

return result.getMessage();

}

/\*

\* 删除平台

\*/

@RequestMapping(value="delePlatform.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String delePlatform(@ModelAttribute("model") PlatformModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = platformService.delePlatform(model);

return result.getMessage();

}

}

package com.sdocean.station.action;

import java.util.ArrayList;

import java.util.List;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.apache.log4j.Logger;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.servlet.ModelAndView;

import com.google.gson.JsonIOException;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.common.service.CommonService;

import com.sdocean.common.service.ZTreeService;

import com.sdocean.company.model.CompanyModel;

import com.sdocean.company.service.CompanyService;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.service.OperationLogService;

import com.sdocean.login.action.LoginAction;

import com.sdocean.page.model.PageResult;

import com.sdocean.page.model.UiColumn;

import com.sdocean.region.model.RegionModel;

import com.sdocean.region.service.RegionService;

import com.sdocean.role.model.RoleModel;

import com.sdocean.station.model.StationModel;

import com.sdocean.station.service.StationService;

import com.sdocean.users.model.SysUser;

@Controller

public class StationAction {

private static Logger log = Logger.getLogger(StationAction.class);

@Resource

StationService stationService;

@Resource

OperationLogService logService;

@Autowired

private ConfigInfo info;

@Autowired

private CompanyService companyService;

@Autowired

private ZTreeService ztreeService;

@Autowired

private RegionService regionService;

@RequestMapping("info\_station.do")

public ModelAndView info\_station(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/station/stationInfo");

return mav;

}

@RequestMapping(value="showStationList.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String showStationTypes(@ModelAttribute("model") StationModel model,HttpServletRequest request,

HttpServletResponse response){

PageResult result = new PageResult();

//为查询结果增加表头

List<UiColumn> cols = stationService.getCols4TypeList();

result.setCols(cols);

//为查询结果增加列表

List<StationModel> rows = stationService.getStationList(model);

result.setRows(rows);

return JsonUtil.toJson(result);

}

/\*

\* 保存修改站点信息

\*/

@RequestMapping(value="saveStaionChange.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveStaionChange(@ModelAttribute("model") StationModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = stationService.saveStaionChange(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

/\*

\* 保存新增站点信息

\*/

@RequestMapping(value="saveNewStation.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveNewStation(@ModelAttribute("model") StationModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = stationService.saveNewStation(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

/\*

\* 获得ZTREE展示列表

\*/

@RequestMapping(value="getStationList4ZTree.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStationList4ZTree(@ModelAttribute("role") RoleModel role,HttpServletRequest request,

HttpServletResponse response){

List<ZTreeModel> result = new ArrayList<ZTreeModel>();

result = stationService.getStationListByRole(role);

return JsonUtil.toJson(result);

}

/\*

\* 获得session中保存的station

\*/

@RequestMapping(value="getStation.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStation(HttpServletRequest request,

HttpServletResponse response){

HttpSession session = request.getSession();

StationModel station = (StationModel) session.getAttribute("station");

return JsonUtil.toJson(station);

}

/\*

\* 获得当前用户权限下的站点列表

\*/

@RequestMapping(value="getStationByUser.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStationByUser(HttpServletRequest request,

HttpServletResponse response){

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

List<StationModel> list = stationService.getStations4User(user);

return JsonUtil.toJson(list);

}

@RequestMapping(value="showStationList4Sms.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String showStationList4Sms(HttpServletRequest request,

HttpServletResponse response){

PageResult result = new PageResult();

//为查询结果增加表头

List<UiColumn> cols = stationService.getStationCols4Sms();

result.setCols(cols);

//为查询结果增加列表

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

List<StationModel> rows = stationService.getStations4User(user);

result.setRows(rows);

return JsonUtil.toJson(result);

}

/\*

\* 获得组织机构下的站点设备列表,

\* 作为载体的参数,不接受跳转

\*/

@RequestMapping(value="getStationDevice4ComanyZtree.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStationDevice4ComanyZtree(HttpServletRequest request,

HttpServletResponse response,ZTreeModel ztree){

String syspath = "http://"+request.getHeader("host")+request.getContextPath();

List<ZTreeModel> ztrees = new ArrayList<ZTreeModel>();

//获得用户权限下的站点的列表

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

List<StationModel> stations = stationService.getStations4User(user);

//根据站点列表获得对应的组织机构列表

List<CompanyModel> comps = new ArrayList<CompanyModel>();

List<ZTreeModel> comztrees = new ArrayList<ZTreeModel>();

List<ZTreeModel> stationztrees = new ArrayList<ZTreeModel>();

List<ZTreeModel> deviceztrees = new ArrayList<ZTreeModel>();

for(StationModel station:stations){

comps.addAll(companyService.getComListByCode(station.getCompanyId()));

List<ZTreeModel> devices = stationService.getDeviceZTree4Station(station);

for(ZTreeModel z:devices){

List<ZTreeModel> cdevices = stationService.getDeviceZTree4Device(z,station.getId());

if(cdevices==null||cdevices.size()<1){

z.setUrl(ztree.getUrl()+z.getId());

}

for(ZTreeModel c:cdevices){

c.setUrl(ztree.getUrl()+c.getId());

}

z.setChildren(cdevices);

}

deviceztrees.addAll(devices);

}

List<CompanyModel> coms = new ArrayList<CompanyModel>();

//将重复的组织机构去掉

for(CompanyModel com:comps){

if(!coms.contains(com)){

coms.add(com);

}

}

//将组织机构列表以及站点列表转换成ztree的形式

try {

comztrees = ztreeService.changeModel2ZTree(coms, "code", "pcode", "shortName",true,"","","",syspath);

stationztrees = ztreeService.changeModel2ZTree(stations, "id", "companyId", "title",true,"url",ZTreeModel.getSelf(),"icon",syspath);

ztrees.addAll(comztrees);

ztrees.addAll(stationztrees);

ztrees.addAll(deviceztrees);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return JsonUtil.toJson(ztrees);

}

/\*

\* 获得组织机构的站点列表

\*/

@RequestMapping(value="getStation4ComanyZtreeByUser.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStation4ComanyZtreeByUser(HttpServletRequest request,

HttpServletResponse response,ZTreeModel ztree){

String syspath = "http://"+request.getHeader("host")+request.getContextPath();

List<ZTreeModel> ztrees = new ArrayList<ZTreeModel>();

//获得用户权限下的站点的列表

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

List<StationModel> stations = stationService.getStations4User(user);

//根据站点列表获得对应的组织机构列表

List<CompanyModel> comps = new ArrayList<CompanyModel>();

List<ZTreeModel> comztrees = new ArrayList<ZTreeModel>();

List<ZTreeModel> stationztrees = new ArrayList<ZTreeModel>();

List<ZTreeModel> deviceztrees = new ArrayList<ZTreeModel>();

for(StationModel station:stations){

comps.addAll(companyService.getComListByCode(station.getCompanyId()));

station.setUrl(ztree.getUrl()+station.getId());

List<ZTreeModel> devices = stationService.getDeviceZTree4Station(station);

for(ZTreeModel z:devices){

List<ZTreeModel> cdevices = stationService.getDeviceZTree4Device(z,station.getId());

z.setChildren(cdevices);

}

deviceztrees.addAll(devices);

}

List<CompanyModel> coms = new ArrayList<CompanyModel>();

//将重复的组织机构去掉

for(CompanyModel com:comps){

if(!coms.contains(com)){

coms.add(com);

}

}

//将组织机构列表以及站点列表转换成ztree的形式

try {

comztrees = ztreeService.changeModel2ZTree(coms, "code", "pcode", "shortName",true,"","","",syspath);

stationztrees = ztreeService.changeModel2ZTree(stations, "id", "companyId", "title",true,"url",ZTreeModel.getSelf(),"icon",syspath);

ztrees.addAll(comztrees);

ztrees.addAll(stationztrees);

ztrees.addAll(deviceztrees);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return JsonUtil.toJson(ztrees);

}

/\*

\* 获得组织机构的站点\1级设备\2级设备的树形菜单

\*/

@RequestMapping(value="getStationDevicesTree.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStationDevicesTree(HttpServletRequest request,

HttpServletResponse response,ZTreeModel ztree){

String syspath = "http://"+request.getHeader("host")+request.getContextPath();

List<ZTreeModel> ztrees = new ArrayList<ZTreeModel>();

//获得用户权限下的站点的列表

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

List<StationModel> stations = stationService.getStations4User(user);

//根据站点列表获得对应的组织机构列表

List<CompanyModel> comps = new ArrayList<CompanyModel>();

List<ZTreeModel> comztrees = new ArrayList<ZTreeModel>();

List<ZTreeModel> stationztrees = new ArrayList<ZTreeModel>();

List<ZTreeModel> deviceztrees = new ArrayList<ZTreeModel>();

for(StationModel station:stations){

comps.addAll(companyService.getComListByCode(station.getCompanyId()));

if(ztree!=null&&ztree.getStationURL()!=null&&ztree.getStationURL().length()>0){

//设置点击站点时的跳转列表

station.setUrl(ztree.getStationURL()+station.getId());

}

List<ZTreeModel> devices = stationService.getDeviceZTree4Station(station);

for(ZTreeModel z:devices){

//设置点击一级设备时的跳转列表

if(ztree!=null&&ztree.getDeviceURL()!=null&&ztree.getDeviceURL().length()>0){

z.setUrl(ztree.getDeviceURL()+z.getId());

}

List<ZTreeModel> cdevices = stationService.getDeviceZTree4Device(z,station.getId());

for(ZTreeModel c:cdevices){

//设置点击二级设备时的跳转列表

if(ztree!=null&&ztree.getDeviceChildURL()!=null&&ztree.getDeviceChildURL().length()>0){

c.setUrl(ztree.getDeviceChildURL()+c.getId());

}

}

z.setChildren(cdevices);

}

deviceztrees.addAll(devices);

}

List<CompanyModel> coms = new ArrayList<CompanyModel>();

//将重复的组织机构去掉

for(CompanyModel com:comps){

if(!coms.contains(com)){

coms.add(com);

}

}

//将组织机构列表以及站点列表转换成ztree的形式

try {

comztrees = ztreeService.changeModel2ZTree(coms, "code", "pcode", "shortName",true,"","","",syspath);

stationztrees = ztreeService.changeModel2ZTree(stations, "id", "companyId", "title",true,"url",ZTreeModel.getSelf(),"icon",syspath);

ztrees.addAll(comztrees);

ztrees.addAll(stationztrees);

ztrees.addAll(deviceztrees);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return JsonUtil.toJson(ztrees);

}

/\*

\* 获得地区树下的站点列表

\*/

@RequestMapping(value="getStation4RegionZtreeByUser.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStation4RegionZtreeByUser(HttpServletRequest request,

HttpServletResponse response,ZTreeModel ztree){

String syspath = "http://"+request.getHeader("host")+request.getContextPath();

List<ZTreeModel> ztrees = new ArrayList<ZTreeModel>();

//获得用户权限下的站点的列表

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

List<StationModel> stations = stationService.getStations4User(user);

//根据站点列表得到对应的地区列表

List<RegionModel> regions = new ArrayList<>();

List<RegionModel> regs = new ArrayList<>();

List<ZTreeModel> deviceztrees = new ArrayList<ZTreeModel>();

for(StationModel station:stations){

regions.addAll(regionService.getRegionListById(station.getRegion\_id()));

station.setUrl(ztree.getUrl()+station.getId());

List<ZTreeModel> devices = stationService.getDeviceZTree4Station(station);

for(ZTreeModel z:devices){

List<ZTreeModel> cdevices = stationService.getDeviceZTree4Device(z,station.getId());

z.setChildren(cdevices);

}

deviceztrees.addAll(devices);

}

for(RegionModel region:regions){

if(!regs.contains(region)){

regs.add(region);

}

}

//将地区机构列表以及站点列表转换成ZTREE的形式

List<ZTreeModel> regztrees = new ArrayList<ZTreeModel>();

List<ZTreeModel> stationztrees = new ArrayList<ZTreeModel>();

try {

regztrees = ztreeService.changeModel2ZTree(regs, "id", "pid", "text",true,"","","",syspath);

stationztrees = ztreeService.changeModel2ZTree(stations, "id", "region\_id", "title",true,"url",ZTreeModel.getSelf(),"icon",syspath);

ztrees.addAll(regztrees);

ztrees.addAll(stationztrees);

ztrees.addAll(deviceztrees);

} catch (Exception e) {

// TODO: handle exception

e.printStackTrace();

}

return JsonUtil.toJson(ztrees);

}

}

package com.sdocean.station.action;

import java.util.ArrayList;

import java.util.List;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.apache.log4j.Logger;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.servlet.ModelAndView;

import com.google.gson.JsonIOException;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.common.service.CommonService;

import com.sdocean.common.service.ZTreeService;

import com.sdocean.company.model.CompanyModel;

import com.sdocean.company.service.CompanyService;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.service.OperationLogService;

import com.sdocean.login.action.LoginAction;

import com.sdocean.page.model.PageResult;

import com.sdocean.page.model.UiColumn;

import com.sdocean.region.model.RegionModel;

import com.sdocean.region.service.RegionService;

import com.sdocean.role.model.RoleModel;

import com.sdocean.station.model.StationModel;

import com.sdocean.station.service.StationService;

import com.sdocean.users.model.SysUser;

@Controller

public class StationAction {

private static Logger log = Logger.getLogger(StationAction.class);

@Resource

StationService stationService;

@Resource

OperationLogService logService;

@Autowired

private ConfigInfo info;

@Autowired

private CompanyService companyService;

@Autowired

private ZTreeService ztreeService;

@Autowired

private RegionService regionService;

@RequestMapping("info\_station.do")

public ModelAndView info\_station(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/station/stationInfo");

return mav;

}

@RequestMapping(value="showStationList.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String showStationTypes(@ModelAttribute("model") StationModel model,HttpServletRequest request,

HttpServletResponse response){

PageResult result = new PageResult();

//为查询结果增加表头

List<UiColumn> cols = stationService.getCols4TypeList();

result.setCols(cols);

//为查询结果增加列表

List<StationModel> rows = stationService.getStationList(model);

result.setRows(rows);

return JsonUtil.toJson(result);

}

/\*

\* 保存修改站点信息

\*/

@RequestMapping(value="saveStaionChange.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveStaionChange(@ModelAttribute("model") StationModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = stationService.saveStaionChange(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

/\*

\* 保存新增站点信息

\*/

@RequestMapping(value="saveNewStation.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveNewStation(@ModelAttribute("model") StationModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = stationService.saveNewStation(model);

logService.saveOperationLog(result,request);

return result.getMessage();

}

/\*

\* 获得ZTREE展示列表

\*/

@RequestMapping(value="getStationList4ZTree.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStationList4ZTree(@ModelAttribute("role") RoleModel role,HttpServletRequest request,

HttpServletResponse response){

List<ZTreeModel> result = new ArrayList<ZTreeModel>();

result = stationService.getStationListByRole(role);

return JsonUtil.toJson(result);

}

/\*

\* 获得session中保存的station

\*/

@RequestMapping(value="getStation.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStation(HttpServletRequest request,

HttpServletResponse response){

HttpSession session = request.getSession();

StationModel station = (StationModel) session.getAttribute("station");

return JsonUtil.toJson(station);

}

/\*

\* 获得当前用户权限下的站点列表

\*/

@RequestMapping(value="getStationByUser.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStationByUser(HttpServletRequest request,

HttpServletResponse response){

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

List<StationModel> list = stationService.getStations4User(user);

return JsonUtil.toJson(list);

}

@RequestMapping(value="showStationList4Sms.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String showStationList4Sms(HttpServletRequest request,

HttpServletResponse response){

PageResult result = new PageResult();

//为查询结果增加表头

List<UiColumn> cols = stationService.getStationCols4Sms();

result.setCols(cols);

//为查询结果增加列表

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

List<StationModel> rows = stationService.getStations4User(user);

result.setRows(rows);

return JsonUtil.toJson(result);

}

/\*

\* 获得组织机构下的站点设备列表,

\* 作为载体的参数,不接受跳转

\*/

@RequestMapping(value="getStationDevice4ComanyZtree.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStationDevice4ComanyZtree(HttpServletRequest request,

HttpServletResponse response,ZTreeModel ztree){

String syspath = "http://"+request.getHeader("host")+request.getContextPath();

List<ZTreeModel> ztrees = new ArrayList<ZTreeModel>();

//获得用户权限下的站点的列表

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

List<StationModel> stations = stationService.getStations4User(user);

//根据站点列表获得对应的组织机构列表

List<CompanyModel> comps = new ArrayList<CompanyModel>();

List<ZTreeModel> comztrees = new ArrayList<ZTreeModel>();

List<ZTreeModel> stationztrees = new ArrayList<ZTreeModel>();

List<ZTreeModel> deviceztrees = new ArrayList<ZTreeModel>();

for(StationModel station:stations){

comps.addAll(companyService.getComListByCode(station.getCompanyId()));

List<ZTreeModel> devices = stationService.getDeviceZTree4Station(station);

for(ZTreeModel z:devices){

List<ZTreeModel> cdevices = stationService.getDeviceZTree4Device(z,station.getId());

if(cdevices==null||cdevices.size()<1){

z.setUrl(ztree.getUrl()+z.getId());

}

for(ZTreeModel c:cdevices){

c.setUrl(ztree.getUrl()+c.getId());

}

z.setChildren(cdevices);

}

deviceztrees.addAll(devices);

}

List<CompanyModel> coms = new ArrayList<CompanyModel>();

//将重复的组织机构去掉

for(CompanyModel com:comps){

if(!coms.contains(com)){

coms.add(com);

}

}

//将组织机构列表以及站点列表转换成ztree的形式

try {

comztrees = ztreeService.changeModel2ZTree(coms, "code", "pcode", "shortName",true,"","","",syspath);

stationztrees = ztreeService.changeModel2ZTree(stations, "id", "companyId", "title",true,"url",ZTreeModel.getSelf(),"icon",syspath);

ztrees.addAll(comztrees);

ztrees.addAll(stationztrees);

ztrees.addAll(deviceztrees);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return JsonUtil.toJson(ztrees);

}

/\*

\* 获得组织机构的站点列表

\*/

@RequestMapping(value="getStation4ComanyZtreeByUser.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStation4ComanyZtreeByUser(HttpServletRequest request,

HttpServletResponse response,ZTreeModel ztree){

String syspath = "http://"+request.getHeader("host")+request.getContextPath();

List<ZTreeModel> ztrees = new ArrayList<ZTreeModel>();

//获得用户权限下的站点的列表

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

List<StationModel> stations = stationService.getStations4User(user);

//根据站点列表获得对应的组织机构列表

List<CompanyModel> comps = new ArrayList<CompanyModel>();

List<ZTreeModel> comztrees = new ArrayList<ZTreeModel>();

List<ZTreeModel> stationztrees = new ArrayList<ZTreeModel>();

List<ZTreeModel> deviceztrees = new ArrayList<ZTreeModel>();

for(StationModel station:stations){

comps.addAll(companyService.getComListByCode(station.getCompanyId()));

station.setUrl(ztree.getUrl()+station.getId());

List<ZTreeModel> devices = stationService.getDeviceZTree4Station(station);

for(ZTreeModel z:devices){

List<ZTreeModel> cdevices = stationService.getDeviceZTree4Device(z,station.getId());

z.setChildren(cdevices);

}

deviceztrees.addAll(devices);

}

List<CompanyModel> coms = new ArrayList<CompanyModel>();

//将重复的组织机构去掉

for(CompanyModel com:comps){

if(!coms.contains(com)){

coms.add(com);

}

}

//将组织机构列表以及站点列表转换成ztree的形式

try {

comztrees = ztreeService.changeModel2ZTree(coms, "code", "pcode", "shortName",true,"","","",syspath);

stationztrees = ztreeService.changeModel2ZTree(stations, "id", "companyId", "title",true,"url",ZTreeModel.getSelf(),"icon",syspath);

ztrees.addAll(comztrees);

ztrees.addAll(stationztrees);

ztrees.addAll(deviceztrees);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return JsonUtil.toJson(ztrees);

}

/\*

\* 获得组织机构的站点\1级设备\2级设备的树形菜单

\*/

@RequestMapping(value="getStationDevicesTree.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStationDevicesTree(HttpServletRequest request,

HttpServletResponse response,ZTreeModel ztree){

String syspath = "http://"+request.getHeader("host")+request.getContextPath();

List<ZTreeModel> ztrees = new ArrayList<ZTreeModel>();

//获得用户权限下的站点的列表

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

List<StationModel> stations = stationService.getStations4User(user);

//根据站点列表获得对应的组织机构列表

List<CompanyModel> comps = new ArrayList<CompanyModel>();

List<ZTreeModel> comztrees = new ArrayList<ZTreeModel>();

List<ZTreeModel> stationztrees = new ArrayList<ZTreeModel>();

List<ZTreeModel> deviceztrees = new ArrayList<ZTreeModel>();

for(StationModel station:stations){

comps.addAll(companyService.getComListByCode(station.getCompanyId()));

if(ztree!=null&&ztree.getStationURL()!=null&&ztree.getStationURL().length()>0){

//设置点击站点时的跳转列表

station.setUrl(ztree.getStationURL()+station.getId());

}

List<ZTreeModel> devices = stationService.getDeviceZTree4Station(station);

for(ZTreeModel z:devices){

//设置点击一级设备时的跳转列表

if(ztree!=null&&ztree.getDeviceURL()!=null&&ztree.getDeviceURL().length()>0){

z.setUrl(ztree.getDeviceURL()+z.getId());

}

List<ZTreeModel> cdevices = stationService.getDeviceZTree4Device(z,station.getId());

for(ZTreeModel c:cdevices){

//设置点击二级设备时的跳转列表

if(ztree!=null&&ztree.getDeviceChildURL()!=null&&ztree.getDeviceChildURL().length()>0){

c.setUrl(ztree.getDeviceChildURL()+c.getId());

}

}

z.setChildren(cdevices);

}

deviceztrees.addAll(devices);

}

List<CompanyModel> coms = new ArrayList<CompanyModel>();

//将重复的组织机构去掉

for(CompanyModel com:comps){

if(!coms.contains(com)){

coms.add(com);

}

}

//将组织机构列表以及站点列表转换成ztree的形式

try {

comztrees = ztreeService.changeModel2ZTree(coms, "code", "pcode", "shortName",true,"","","",syspath);

stationztrees = ztreeService.changeModel2ZTree(stations, "id", "companyId", "title",true,"url",ZTreeModel.getSelf(),"icon",syspath);

ztrees.addAll(comztrees);

ztrees.addAll(stationztrees);

ztrees.addAll(deviceztrees);

} catch (Exception e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

return JsonUtil.toJson(ztrees);

}

/\*

\* 获得地区树下的站点列表

\*/

@RequestMapping(value="getStation4RegionZtreeByUser.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStation4RegionZtreeByUser(HttpServletRequest request,

HttpServletResponse response,ZTreeModel ztree){

String syspath = "http://"+request.getHeader("host")+request.getContextPath();

List<ZTreeModel> ztrees = new ArrayList<ZTreeModel>();

//获得用户权限下的站点的列表

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

List<StationModel> stations = stationService.getStations4User(user);

//根据站点列表得到对应的地区列表

List<RegionModel> regions = new ArrayList<>();

List<RegionModel> regs = new ArrayList<>();

List<ZTreeModel> deviceztrees = new ArrayList<ZTreeModel>();

for(StationModel station:stations){

regions.addAll(regionService.getRegionListById(station.getRegion\_id()));

station.setUrl(ztree.getUrl()+station.getId());

List<ZTreeModel> devices = stationService.getDeviceZTree4Station(station);

for(ZTreeModel z:devices){

List<ZTreeModel> cdevices = stationService.getDeviceZTree4Device(z,station.getId());

z.setChildren(cdevices);

}

deviceztrees.addAll(devices);

}

for(RegionModel region:regions){

if(!regs.contains(region)){

regs.add(region);

}

}

//将地区机构列表以及站点列表转换成ZTREE的形式

List<ZTreeModel> regztrees = new ArrayList<ZTreeModel>();

List<ZTreeModel> stationztrees = new ArrayList<ZTreeModel>();

try {

regztrees = ztreeService.changeModel2ZTree(regs, "id", "pid", "text",true,"","","",syspath);

stationztrees = ztreeService.changeModel2ZTree(stations, "id", "region\_id", "title",true,"url",ZTreeModel.getSelf(),"icon",syspath);

ztrees.addAll(regztrees);

ztrees.addAll(stationztrees);

ztrees.addAll(deviceztrees);

} catch (Exception e) {

// TODO: handle exception

e.printStackTrace();

}

return JsonUtil.toJson(ztrees);

}

}

package com.sdocean.station.action;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.servlet.ModelAndView;

import com.sdocean.common.model.Result;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.service.OperationLogService;

import com.sdocean.page.model.PageResult;

import com.sdocean.page.model.UiColumn;

import com.sdocean.station.model.StationDeviceModel;

import com.sdocean.station.model.StationModel;

import com.sdocean.station.service.StationDeviceService;

@Controller

public class StationDeviceAction {

@Resource

StationDeviceService stationDeviceService;

@Resource

OperationLogService logService;

@Autowired

private ConfigInfo info;

@RequestMapping("info\_stationDevice.do")

public ModelAndView info\_station(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/station/stationDeviceInfo");

return mav;

}

/\*

\* 展示站点设备配置列表

\*/

@RequestMapping(value="showStationDeviceList.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String showStationDeviceList(@ModelAttribute("model") StationDeviceModel model,HttpServletRequest request,

HttpServletResponse response){

//获得该用户的站点权限

HttpSession session = request.getSession();

List<StationModel> stations = (List<StationModel>) session.getAttribute("stations");

PageResult page = new PageResult();

List<UiColumn> cols = stationDeviceService.getCols4StationDeviceList();

List<StationDeviceModel> rows = stationDeviceService.getStationDeviceList(model,stations);

page.setCols(cols);

page.setRows(rows);

return JsonUtil.toJson(page);

}

/\*

\* 保存新增的站点设备配置

\*/

@RequestMapping(value="saveNewStationDevice.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveNewStationDevice(@ModelAttribute("model") StationDeviceModel model,HttpServletRequest request,

HttpServletResponse response){

//判断创建时间

if(model!=null&&(model.getCreateTime()==null||model.getCreateTime().length()<1)){

Date now = new Date();

SimpleDateFormat sdf=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

model.setCreateTime(sdf.format(now));

}

Result result = stationDeviceService.saveNewStationDevice(model);

//保存操作日志

logService.saveOperationLog(result, request);

return result.getMessage();

}

/\*

\* 保存修改的站点设备配置

\*/

@RequestMapping(value="saveChangeStationDevice.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveChangeStationDevice(@ModelAttribute("model") StationDeviceModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = stationDeviceService.saveChangeStationDevice(model);

//保存操作日志

logService.saveOperationLog(result, request);

return result.getMessage();

}

/\*

\* 删除站点设备配置

\*/

@RequestMapping(value="deleteStationDevice.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String deleteStationDevice(@ModelAttribute("model") StationDeviceModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = stationDeviceService.deleteStationDevice(model);

//保存操作日志

logService.saveOperationLog(result, request);

return result.getMessage();

}

/\*

\* 根据站点ID，设备ID或的站点设备配置

\*/

@RequestMapping(value="getStationDeviceByWidDid.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStationDeviceByWidDid(@ModelAttribute("model") StationDeviceModel model,HttpServletRequest request,

HttpServletResponse response){

StationDeviceModel result = stationDeviceService.getStationDeviceByWidDid(model);

return JsonUtil.toJson(result);

}

}

package com.sdocean.station.action;

import java.util.ArrayList;

import java.util.List;

import javax.annotation.Resource;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RequestMethod;

import org.springframework.web.bind.annotation.ResponseBody;

import org.springframework.web.servlet.ModelAndView;

import com.sdocean.common.model.Result;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.service.OperationLogService;

import com.sdocean.page.model.PageResult;

import com.sdocean.page.model.UiColumn;

import com.sdocean.station.model.StationDeviceIndicator;

import com.sdocean.station.model.StationModel;

import com.sdocean.station.service.StationDeviceIndicatorService;

import com.sdocean.users.model.SysUser;

@Controller

public class StationDeviceIndicatorAction {

@Resource

StationDeviceIndicatorService sdiService;

@Resource

OperationLogService logService;

@Autowired

private ConfigInfo info;

@RequestMapping("info\_stationIndicator.do")

public ModelAndView info\_station(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/station/stationIndicator");

return mav;

}

@RequestMapping(value="getDeiviceIndicatorsByStation.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getDeiviceIndicatorsByStation(@ModelAttribute("model") StationDeviceIndicator model,HttpServletRequest request,

HttpServletResponse response){

List<ZTreeModel> list = new ArrayList<ZTreeModel>();

list = sdiService.getDeiviceIndicatorsByStation(model.getStationId());

return JsonUtil.toJson(list);

}

/\*

\* 获得用户权限下的站点的列表

\*/

@RequestMapping(value="getStation4DeviceIndicator.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getStation4DeviceIndicator(@ModelAttribute("model") StationDeviceIndicator model,HttpServletRequest request,

HttpServletResponse response){

PageResult page = new PageResult();

List<UiColumn> cols = sdiService.getCols4Stations();

List<StationModel> stations = new ArrayList<StationModel>();

HttpSession session = request.getSession();

SysUser user = (SysUser) session.getAttribute("user");

stations = sdiService.getStation4DeviceIndicator(user.getId(), model.getStationName());

page.setCols(cols);

page.setRows(stations);

return JsonUtil.toJson(page);

}

/\*

\* 保存站点设备参数

\*/

@RequestMapping(value="saveStationDeviceIndicator.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveStationDeviceIndicator(@ModelAttribute("model") StationDeviceIndicator model,HttpServletRequest request,

HttpServletResponse response){

Result result = sdiService.saveStationDeviceIndicator(model);

logService.saveOperationLog(result, request);

return JsonUtil.toJson(result.getMessage());

}

}

<%@ page contentType="text/html; charset=UTF-8"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn"%>

<!DOCTYPE html>

<html >

<c:set var="ctx" value="${pageContext.request.contextPath}"/>

<head>

<meta charset="utf-8" />

<title>${system.systemName }</title>

<meta name="keywords" content="入海污染源" />

<script language="javascript">

var dooropenfunction= function (){

if(document.getElementById("door\_left").width>2&&document.getElementById("menzhuangtai").options[document.getElementById("menzhuangtai").selectedIndex].value=="kai"){

document.getElementById("door\_left").width-=2;

document.getElementById("door\_right").width-=2;

}else{

dooropenfunction=null;

}

}

var doorclosfunction = function(){

if(document.getElementById("door\_left").width<26&&document.getElementById("menzhuangtai").options[document.getElementById("menzhuangtai").selectedIndex].value=="guan"){

document.getElementById("door\_left").width+=2;

document.getElementById("door\_right").width+=2;

}

else{

doorclosfunction=null;

}

}

function showdoor(){

if(document.getElementById("menzhuangtai").options[document.getElementById("menzhuangtai").selectedIndex].value=="kai"){

window.setInterval(dooropenfunction, 200);

}else if(document.getElementById("menzhuangtai").options[document.getElementById("menzhuangtai").selectedIndex].value=="guan"){

window.setInterval(doorclosfunction, 200);

}

}

</script>

<style>

.kongzhimianban-imgnew{

width:250px;

height:230px;

background-image:url(img/donghua1.jpg);

background-size: 250px 230px;

margin-left: 10px;

margin-bottom: -5px;

}

.kongzhimianban-img{

width:250px;

height:230px;

background-image:url(img/donghua2.jpg);

background-size: 250px 230px;

margin-left: 10px;

margin-bottom: -5px;

}

#door\_left{

margin-top: 66px;

margin-left: 146px;

}

#door\_right{

margin-top: 66px;

margin-left:-3px;

float: right;

margin-right: 20px;

}

#demo {

background: #FFF;

overflow:hidden;

border: 0;

width: 20px;

height:8px;

margin-left: 110px;

margin-top: -2px;

}

#indemo {

float: left;

width: 800%;

}

#demo1 {

float: left;

}

#demo2 {

float: left;

}

#demo1 img{

margin-top:-11px;

height:8px;

width:20px;

}

#demo2 img{

margin-top:-4px;

}

</style>

</head>

<body ng-app="myApp" ng-controller="customersCtrl">

<%@ include file="../common/top.jsp" %>

<link rel="stylesheet" href="${ctx }/resources/angular-select-tree/angular-multi-select-tree-0.1.0.css" />

<script type="text/javascript" src="${ctx }/resources/angular-select-tree/angular-multi-select-tree-0.1.0.js"></script>

<script type="text/javascript" src="${ctx }/resources/angular-select-tree/angular-multi-select-tree-0.1.0.tpl.js"></script>

<script src="${ctx }/manage/js/device/deviceInfo.js"></script>

<div class="main-container" id="main-container">

<script type="text/javascript">

try{ace.settings.check('main-container' , 'fixed')}catch(e){}

</script>

<div class="main-container-inner">

<%@ include file="../common/leftMenu4device.jsp" %>

<div class="main-content" >

<div class="breadcrumbs" id="breadcrumbs">

<script type="text/javascript">

try{ace.settings.check('breadcrumbs' , 'fixed')}catch(e){}

</script>

<ul class="breadcrumb">

<li>

<i class="icon-home home-icon"></i>

<a href="#">${currMenu.pMenuName }</a>

</li>

<li>

<a href="#">${currMenu.cMenuName }</a>

</li>

</ul><!-- .breadcrumb -->

</div>

<div class="container">

<div class="row">

<div class="col-md-12 col-sm-12 col-sx-12 map-main">

<div class="panel panel-default">

<div class="panel-body panel-body-margin" >

<div class="row">

<div class="col-md-9 col-sm-9 map-right ">

<div class="panel panel-default panel-default-right">

<ul id="myTab" class="nav nav-tabs nav-tabs-tongji ">

<li class="">

<a href="#activites" data-toggle="tab" data-original-title="">自动控制</a>

</li>

<li class="active">

<a href="#signups" data-toggle="tab" data-original-title="">手动控制</a>

</li>

</ul>

<div class="panel-body panel-body-tongji">

<section class="todo">

<fieldset class="todo-list">

<div id="myTabContent" class="tab-content">

<div class="tab-pane fade" id="activites">

<div class="row">

<div class="col-md-8 col-sm-8 kongzhi-height-left">

<div class="panel panel-default kongzhi-panel-default">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">控制面板</h3>

</div>

<div class="panel-body">

<div class="row">

<div class="col-md-7 row-kongzhimianban kongzhimianban-img ">

</div>

<div class="col-md-5 row-kongzhimianban">

<div class="input-group ">

<span class="input-group-addon input-group-addon-control ">进水泵</span>

<select class="form-control input-sm" disabled>

<option value="">浅层</option>

<option value="">深层</option>

</select>

</div>

<div class="input-group">

<span class="input-group-addon input-group-addon-control">泵方向</span>

<select class="form-control input-sm " disabled>

<option value="">进水</option>

<option value="">出水</option>

</select>

</div>

<div class="input-group">

<span class="input-group-addon input-group-addon-control">门状态</span>

<select class="form-control input-sm " disabled>

<option value="">开</option>

<option value="">关</option>

</select>

</div>

<div class="input-group ">

<span class="input-group-addon input-group-addon-control " >泵控制</span>

<select class="form-control input-sm" disabled>

<option value="">开</option>

<option value="">关</option>

</select>

</div>

<div class="input-group">

<span class="input-group-addon input-group-addon-control">油电源</span>

<select class="form-control input-sm " disabled>

<option value="">开</option>

<option value="">关</option>

</select>

</div>

</div>

<div class="col-md-12">

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="" disabled>潜水管清洗</button>

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="" disabled>深水管清洗</button>

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="" disabled>排空气</button>

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="" disabled>校准</button>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="col-md-4 col-sm-4 kongzhi-height-right">

<div class="panel panel-default kongzhi-panel-default">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">时间间隔参数</h3>

</div>

<div class="panel-body">

<div class="input-group ">

<span class="input-group-addon ">物理参数</span>

<select class="form-control input-sm" disabled>

<option value="">20min</option>

<option value="">30min</option>

</select>

</div>

<div class="input-group ">

<span class="input-group-addon ">营养盐</span>

<select class="form-control input-sm" disabled>

<option value="">20min</option>

<option value="">30min</option>

</select>

</div>

<div class="table-responsive ">

<table class="table no-margin">

<thead>

<tr>

<th>#</th>

<th>排水</th>

<th>进水</th>

<th>进水选择</th>

</tr>

</thead>

<tbody>

<tr>

<td>浅层</td>

<td><input class="form-control form-control-kongzhi" type="text"></td>

<td><input class="form-control form-control-kongzhi" type="text"></td>

<td><input class="form-control form-control-kongzhi" type="text"></td>

</tr>

<tr>

<td>深层</td>

<td><input class="form-control form-control-kongzhi" type="text"></td>

<td><input class="form-control form-control-kongzhi" type="text"></td>

<td><input class="form-control form-control-kongzhi" type="text"></td>

</tr>

</tbody>

</table>

</div>

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="">更新</button>

</div>

</div>

</div>

</div>

<div class="col-md-12">

<div class="panel-body center-align-text">

<span >数据浏览</span>

<select >

<option value="">浅层</option>

<option value="">深层</option>

</select>

<label class="checkbox-inline">

<input class="todo-list-cb" type="checkbox">

<span class="todo-list-mark"></span>

<span class="todo-list-desc">更新水质探头</span>

</label>

<label class="checkbox-inline">

<input class="todo-list-cb" type="checkbox">

<span class="todo-list-mark"></span>

<span class="todo-list-desc">更新油探头</span>

</label>

<label class="checkbox-inline">

<input class="todo-list-cb" type="checkbox">

<span class="todo-list-mark"></span>

<span class="todo-list-desc">更新营养盐标准品</span>

</label>

<label class="checkbox-inline">

<input class="todo-list-cb" type="checkbox">

<span class="todo-list-mark"></span>

<span class="todo-list-desc">更新营养盐样品</span>

</label>

<button type="button" class="btn btn-xs btn-default pilianggengxin " data-original-title="" disabled>自动批量更新</button>

</div>

</div>

<div class="col-md-12">

<table>

<tbody><tr>

<td>

<div class="panel panel-default panel-default-table">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">水质参数</h3>

</div>

<div class="panel-body">

<table>

<tbody>

<tr>

<td>蓝绿藻：</td>

<td>XXXX</td>

</tr>

<tr>

<td>叶绿素：</td>

<td>XXXX</td>

</tr>

<tr>

<td>电导率：</td>

<td>XXXX</td>

</tr>

<tr>

<td>溶解氧</td>

<td>XXXX</td>

</tr>

<tr>

<td>PH值：</td>

<td>XXXX</td>

</tr>

<tr>

<td>温度：</td>

<td>XXXX</td>

</tr>

<tr>

<td>浊度：</td>

<td>XXXX</td>

</tr>

<tr>

<td colspan="2"> 上次更新时间：</td>

</tr>

<tr>

<td colspan="2"> 2014-3-2 12:00:00</td>

</tr>

</tbody>

</table>

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="" disabled>手动更新</button>

</div>

</div>

</div>

</td>

<td>

<div class="panel panel-default panel-default-table">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">油参数</h3>

</div>

<div class="panel-body canshu-table">

<table>

<tbody>

<tr>

<td>原油：</td>

<td>16.9</td>

</tr>

<tr>

<td>成品油：</td>

<td>79.9</td>

</tr>

<tr>

<td>温度：</td>

<td>5.6.C</td>

</tr>

<tr>

<td></td>

<td></td>

</tr>

<tr>

<td></td>

<td></td>

</tr>

<tr>

<td></td>

<td></td>

</tr>

<tr>

<td></td>

<td></td>

</tr>

<tr>

<td colspan="2"> 上次更新时间：</td>

</tr>

<tr>

<td colspan="2"> 2014-3-2 12:00:00</td>

</tr>

</tbody>

</table>

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="" disabled>手动更新</button>

</div>

</div>

</div>

</td>

<td>

<div class="panel panel-default panel-default-table">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">营养盐标准品</h3>

</div>

<div class="panel-body canshu-table">

<table>

<tbody>

<tr>

<td>硝酸盐：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td>磷酸盐：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td>氨：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td>硅酸盐：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

</tr>

<tr>

<td colspan="2"> 上次更新时间：</td>

</tr>

<tr>

<td colspan="2"> 2014-3-2 12:00:00</td>

</tr>

</tbody>

</table>

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="" disabled>手动更新</button>

</div>

</div>

</div>

</td>

<td>

<div class="panel panel-default panel-default-table">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">营养盐样品</h3>

</div>

<div class="panel-body canshu-table">

<table>

<tbody><tr>

<td>硝酸盐：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td>磷酸盐：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td>氨：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td>硅酸盐：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

</tr>

<tr>

</tr><tr>

<td colspan="2"> 上次更新时间：</td>

</tr>

<tr>

<td colspan="2"> 2014-3-2 12:00:00</td>

</tr>

</tbody>

</table>

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="" disabled>手动更新</button>

</div>

</div>

</div>

</td>

<td>

<div class="panel panel-default panel-default-table">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">天气参数</h3>

</div>

<div class="panel-body canshu-table">

<table>

<tbody>

<tr>

<td>空气温度：</td>

<td>23</td>

</tr>

<tr>

<td>太阳辐照度：</td>

<td>22</td>

</tr>

<tr>

<td>雨量：</td>

<td>22ml</td>

</tr>

<tr>

<td>相对湿度：</td>

<td>3</td>

</tr>

<tr>

<td>电池电压：</td>

<td>12.6</td>

</tr>

<tr>

<td></td>

<td></td>

</tr>

<tr>

<td></td>

<td></td>

</tr>

<tr>

<td colspan="2"> 上次更新时间：</td>

</tr>

<tr>

<td colspan="2"> 2014-3-2 12:00:00</td>

</tr>

</tbody>

</table>

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="" disabled>手动更新</button>

</div>

</div>

</div>

</td>

</tr>

</tbody></table>

</div>

</div>

</div>

<div class="tab-pane fade active in" id="signups">

<div class="row">

<div class="col-md-8 col-sm-8 kongzhi-height-left">

<div class="panel panel-default kongzhi-panel-default">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">控制面板</h3>

</div>

<div class="panel-body">

<div class="row">

<div id="kongzhiimg" class="col-md-7 row-kongzhimianban kongzhimianban-img">

<img src="images/device/door-left.png" id="door\_left" width="27px" height="35px" />

<img src="images/device/door-right.png" id="door\_right" width="27px" height="35px" />

<div id="demo">

<div id="indemo">

<div id="demo1">

<img src="../live/img/dianji1.png" border="0" />

</div>

<div id="demo2"></div>

</div>

</div>

<script>

var speed=10;

var tab=document.getElementById("demo");

var tab1=document.getElementById("demo1");

var tab2=document.getElementById("demo2");

tab2.innerHTML=tab1.innerHTML;

var fadongjistart=null;

fadongjistart =function (){

if(document.getElementById("bengkongzhi").options[document.getElementById("bengkongzhi").selectedIndex].value=="kai"){

if(tab2.offsetWidth-tab.scrollLeft<=0)

{

tab.scrollLeft-=tab1.offsetWidth

}

else{

tab.scrollLeft++;

}

}

}

function begin(){

if(document.getElementById("bengkongzhi").options[document.getElementById("bengkongzhi").selectedIndex].value=="guan"){

fadongjistart=null;

}

window.setInterval(fadongjistart,speed);

}

function bengfangxiang(){

if(document.getElementById("bengkaiguan").options[document.getElementById("bengkaiguan").selectedIndex].value=="jinshui"){

var str=document.getElementById( "kongzhiimg").className;

document.getElementById( "kongzhiimg").className=str.substring(0,str.length-3);

}else if(document.getElementById("bengkaiguan").options[document.getElementById("bengkaiguan").selectedIndex].value=="chushui"){

document.getElementById( "kongzhiimg").className=document.getElementById( "kongzhiimg").className+"new";

}

}

</script>

</div>

<div class="col-md-5 row-kongzhimianban">

<div class="input-group ">

<span class="input-group-addon input-group-addon-control ">进水泵</span>

<select class="form-control input-sm">

<option value="">浅层</option>

</select>

</div>

<div class="input-group">

<span class="input-group-addon input-group-addon-control">泵方向</span>

<select id="bengkaiguan" onChange="bengfangxiang()" class="form-control input-sm">

<option value="jinshui">进水</option>

<option value="chushui">出水</option>

</select>

</div>

<div class="input-group">

<span class="input-group-addon input-group-addon-control">门状态</span>

<select id="menzhuangtai" onChange="showdoor()" class="form-control input-sm input-sm">

<option value="guan">关</option>

<option value="kai">开</option>

</select>

</div>

<div class="input-group ">

<span class="input-group-addon input-group-addon-control ">泵控制</span>

<select id="bengkongzhi" onChange="begin()" class="form-control input-sm">

<option value="guan">关</option>

<option value="kai">开</option>

</select>

</div>

<div class="input-group">

<span class="input-group-addon input-group-addon-control">油电源</span>

<select class="form-control input-sm">

<option value="">开</option>

<option value="">关</option>

</select>

</div>

</div>

<div class="col-md-12">

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="">潜水管清洗</button>

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="">深水管清洗</button>

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="">排空气</button>

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="">校准</button>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="col-md-4 col-sm-4 kongzhi-height-right">

<div class="panel panel-default kongzhi-panel-default">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">时间间隔参数</h3>

</div>

<div class="panel-body">

<div class="input-group ">

<span class="input-group-addon ">物理参数</span>

<select class="form-control input-sm">

<option value="">20min</option>

<option value="">30min</option>

</select>

</div>

<div class="input-group ">

<span class="input-group-addon ">营养盐</span>

<select class="form-control input-sm">

<option value="">20min</option>

<option value="">30min</option>

</select>

</div>

<div class="table-responsive ">

<table class="table no-margin">

<thead>

<tr>

<th>#</th>

<th>排水</th>

<th>进水</th>

<th>进水选择</th>

</tr>

</thead>

<tbody>

<tr>

<td>浅层</td>

<td><input class="form-control form-control-kongzhi" type="text"></td>

<td><input class="form-control form-control-kongzhi" type="text"></td>

<td><input class="form-control form-control-kongzhi" type="text"></td>

</tr>

<tr>

<td>深层</td>

<td><input class="form-control form-control-kongzhi" type="text"></td>

<td><input class="form-control form-control-kongzhi" type="text"></td>

<td><input class="form-control form-control-kongzhi" type="text"></td>

</tr>

</tbody>

</table>

</div>

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="">更新</button>

</div>

</div>

</div>

</div>

<div class="col-md-12">

<div class="panel-body center-align-text">

<span >数据浏览</span>

<select >

<option value="">浅层</option>

<option value="">深层</option>

</select>

<label class="checkbox-inline">

<input class="todo-list-cb" type="checkbox">

<span class="todo-list-mark"></span>

<span class="todo-list-desc">更新水质探头</span>

</label>

<label class="checkbox-inline">

<input class="todo-list-cb" type="checkbox">

<span class="todo-list-mark"></span>

<span class="todo-list-desc">更新油探头</span>

</label>

<label class="checkbox-inline">

<input class="todo-list-cb" type="checkbox">

<span class="todo-list-mark"></span>

<span class="todo-list-desc">更新营养盐标准品</span>

</label>

<label class="checkbox-inline">

<input class="todo-list-cb" type="checkbox">

<span class="todo-list-mark"></span>

<span class="todo-list-desc">更新营养盐样品</span>

</label>

<button type="button" class="btn btn-xs btn-default pilianggengxin" data-original-title="">自动批量更新</button>

</div>

</div>

<div class="col-md-12" >

<table >

<tr>

<td>

<div class="panel panel-default panel-default-table">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">水质参数</h3>

</div>

<div class="panel-body">

<table >

<tbody>

<tr>

<td>蓝绿藻：</td>

<td>XXXX</td>

</tr>

<tr>

<td>叶绿素：</td>

<td>XXXX</td>

</tr>

<tr>

<td>电导率：</td>

<td>XXXX</td>

</tr>

<tr>

<td>溶解氧</td>

<td>XXXX</td>

</tr>

<tr>

<td>PH值：</td>

<td>XXXX</td>

</tr>

<tr>

<td>温度：</td>

<td>XXXX</td>

</tr>

<tr>

<td>浊度：</td>

<td>XXXX</td>

</tr>

<tr>

<td colspan="2"> 上次更新时间：</td>

</tr>

<tr>

<td colspan="2"> 2014-3-2 12:00:00</td>

</tr>

</tbody>

</table>

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="">手动更新</button>

</div>

</div>

</div>

</td>

<td>

<div class="panel panel-default panel-default-table">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">油参数</h3>

</div>

<div class="panel-body canshu-table">

<table >

<tbody>

<tr>

<td>原油：</td>

<td>16.9</td>

</tr>

<tr>

<td>成品油：</td>

<td>79.9</td>

</tr>

<tr>

<td>温度：</td>

<td>5.6.C</td>

</tr>

<tr>

<td></td>

<td></td>

</tr>

<tr>

<td></td>

<td></td>

</tr>

<tr>

<td></td>

<td></td>

</tr>

<tr>

<td></td>

<td></td>

</tr>

<tr>

<td colspan="2"> 上次更新时间：</td>

</tr>

<tr>

<td colspan="2"> 2014-3-2 12:00:00</td>

</tr>

</tbody>

</table>

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="">手动更新</button>

</div>

</div>

</div>

</td>

<td>

<div class="panel panel-default panel-default-table">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">营养盐标准品</h3>

</div>

<div class="panel-body canshu-table">

<table >

<tbody>

<tr>

<td>硝酸盐：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td>磷酸盐：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td>氨：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td>硅酸盐：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

</tr>

<tr>

<td colspan="2"> 上次更新时间：</td>

</tr>

<tr>

<td colspan="2"> 2014-3-2 12:00:00</td>

</tr>

</tbody>

</table>

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="">手动更新</button>

</div>

</div>

</div>

</td>

<td>

<div class="panel panel-default panel-default-table">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">营养盐样品</h3>

</div>

<div class="panel-body canshu-table">

<table >

<tr>

<td>硝酸盐：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td>磷酸盐：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td>氨：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td>硅酸盐：</td>

<td>0.9MG/L</td>

</tr>

<tr>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

</tr>

<tr>

<td> </td>

<td> </td>

</tr>

<tr>

<tr>

<td colspan="2"> 上次更新时间：</td>

</tr>

<tr>

<td colspan="2"> 2014-3-2 12:00:00</td>

</tr>

</tbody>

</table>

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="">手动更新</button>

</div>

</div>

</div>

</td>

<td>

<div class="panel panel-default panel-default-table">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">天气参数</h3>

</div>

<div class="panel-body canshu-table">

<table >

<tbody>

<tr>

<td>空气温度：</td>

<td>23</td>

</tr>

<tr>

<td>太阳辐照度：</td>

<td>22</td>

</tr>

<tr>

<td>雨量：</td>

<td>22ml</td>

</tr>

<tr>

<td>相对湿度：</td>

<td>3</td>

</tr>

<tr>

<td>电池电压：</td>

<td>12.6</td>

</tr>

<tr>

<td></td>

<td></td>

</tr>

<tr>

<td></td>

<td></td>

</tr>

<tr>

<td colspan="2"> 上次更新时间：</td>

</tr>

<tr>

<td colspan="2"> 2014-3-2 12:00:00</td>

</tr>

</tbody>

</table>

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="">手动更新</button>

</div>

</div>

</div>

</td>

</tr>

</table>

</div>

</div>

</div>

</div>

</fieldset>

</section>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="panel-footer clearfix">

</div>

</div>

</div>

</div>

</div>

</div>

</body>

<script language="javascript">

var dooropenfunction= function (){

if(document.getElementById("door\_left").width>2&&document.getElementById("menzhuangtai").options[document.getElementById("menzhuangtai").selectedIndex].value=="kai"){

document.getElementById("door\_left").width-=2;

document.getElementById("door\_right").width-=2;

}else{

dooropenfunction=null;

}

}

var doorclosfunction = function(){

if(document.getElementById("door\_left").width<26&&document.getElementById("menzhuangtai").options[document.getElementById("menzhuangtai").selectedIndex].value=="guan"){

document.getElementById("door\_left").width+=2;

document.getElementById("door\_right").width+=2;

}

else{

doorclosfunction=null;

}

}

function showdoor(){

if(document.getElementById("menzhuangtai").options[document.getElementById("menzhuangtai").selectedIndex].value=="kai"){

window.setInterval(dooropenfunction, 200);

}else if(document.getElementById("menzhuangtai").options[document.getElementById("menzhuangtai").selectedIndex].value=="guan"){

window.setInterval(doorclosfunction, 200);

}

}

</script>

<style>

.kongzhimianban-imgnew{

width:250px;

height:230px;

background-image:url(img/donghua1.jpg);

background-size: 250px 230px;

margin-left: 10px;

margin-bottom: -5px;

}

.kongzhimianban-img{

width:250px;

height:230px;

background-image:url(${ctx }/images/device/donghua2.jpg);

background-size: 250px 230px;

margin-left: 10px;

margin-bottom: -5px;

}

.kongzhimianban-img2{

width:250px;

height:230px;

background-image:url(${ctx }/images/device/shexiangtou.png);

background-size: 250px 230px;

margin-left: 10px;

margin-bottom: -5px;

}

#door\_left{

margin-top: 66px;

margin-left: 146px;

}

#door\_right{

margin-top: 66px;

margin-left:-3px;

float: right;

margin-right: 20px;

}

#demo {

background: #FFF;

overflow:hidden;

border: 0;

width: 20px;

height:8px;

margin-left: 110px;

margin-top: -2px;

}

#indemo {

float: left;

width: 800%;

}

#demo1 {

float: left;

}

#demo2 {

float: left;

}

#demo1 img{

margin-top:-11px;

height:8px;

width:20px;

}

#demo2 img{

margin-top:-4px;

}

</style>

</html>

<%@ page contentType="text/html; charset=UTF-8"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt"%>

<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn"%>

<!DOCTYPE html>

<html >

<c:set var="ctx" value="${pageContext.request.contextPath}"/>

<head>

<meta charset="utf-8" />

<title>${system.systemName }</title>

<meta name="keywords" content="入海污染源" />

</head>

<body ng-app="myApp" ng-controller="customersCtrl">

<%@ include file="../common/top.jsp" %>

<link rel="stylesheet" href="${ctx }/resources/angular-select-tree/angular-multi-select-tree-0.1.0.css" />

<script type="text/javascript" src="${ctx }/resources/angular-select-tree/angular-multi-select-tree-0.1.0.js"></script>

<script type="text/javascript" src="${ctx }/resources/angular-select-tree/angular-multi-select-tree-0.1.0.tpl.js"></script>

<script src="${ctx }/manage/js/device/deviceInfo.js"></script>

<div class="main-container" id="main-container">

<script type="text/javascript">

try{ace.settings.check('main-container' , 'fixed')}catch(e){}

</script>

<div class="main-container-inner">

<%@ include file="../common/leftMenu4device.jsp" %>

<div class="main-content" >

<div class="breadcrumbs" id="breadcrumbs">

<script type="text/javascript">

try{ace.settings.check('breadcrumbs' , 'fixed')}catch(e){}

</script>

<ul class="breadcrumb">

<li>

<i class="icon-home home-icon"></i>

<a href="#">${currMenu.pMenuName }</a>

</li>

<li>

<a href="#">${currMenu.cMenuName }</a>

</li>

</ul><!-- .breadcrumb -->

</div>

<div class="col-md-12 col-sm-12 col-sx-12 map-main">

<div class="panel panel-default">

<div class="row">

<div class="col-md-11 col-sm-11 map-right ">

<section class="todo">

<fieldset class="todo-list">

<div class="tab-pane fade active in" id="signups">

<div class="row">

<div class="col-md-8 col-sm-8 kongzhi-height-left">

<div class="panel panel-default kongzhi-panel-default">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">潜标</h3>

</div>

<div class="panel-body">

<div class="row">

<div id="kongzhiimg" class="col-md-7 row-kongzhimianban kongzhimianban-img">

</div>

<div class="col-md-5 row-kongzhimianban">

<div class="input-group ">

<span class="input-group-addon input-group-addon-control ">电量</span>

<select class="form-control input-sm">

<option value="">正常</option>

</select>

</div>

<div class="input-group">

<span class="input-group-addon input-group-addon-control">电压</span>

<select id="bengkaiguan" onChange="bengfangxiang()" class="form-control input-sm">

<option value="chushui">12V</option>

</select>

</div>

<div class="input-group">

<span class="input-group-addon input-group-addon-control">急停</span>

<select id="menzhuangtai" onChange="showdoor()" class="form-control input-sm input-sm">

<option value="guan">关</option>

</select>

</div>

<div class="input-group ">

<span class="input-group-addon input-group-addon-control ">姿态</span>

<select id="bengkongzhi" onChange="begin()" class="form-control input-sm">

<option value="guan">正常</option>

</select>

</div>

<div class="input-group">

<span class="input-group-addon input-group-addon-control">运行</span>

<select class="form-control input-sm">

<option value="">正常</option>

</select>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</fieldset>

</section>

</div>

</div>

</div>

</div>

<div class="col-md-12 col-sm-12 col-sx-12 map-main">

<div class="panel panel-default">

<div class="row">

</div>

</div>

</div>

<div class="col-md-12 col-sm-12 col-sx-12 map-main">

<div class="panel panel-default">

<div class="row">

<div class="col-md-11 col-sm-11 map-right ">

<section class="todo">

<fieldset class="todo-list">

<div class="tab-pane fade active in" id="signups">

<div class="row">

<div class="col-md-8 col-sm-8 kongzhi-height-left">

<div class="panel panel-default kongzhi-panel-default">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">CTD</h3>

</div>

<div class="panel-body">

<div class="row">

<div id="kongzhiimg" class="col-md-7 row-kongzhimianban kongzhimianban-img2">

</div>

<div class="col-md-5 row-kongzhimianban">

<div class="input-group ">

<span class="input-group-addon input-group-addon-control ">电压</span>

<select class="form-control input-sm">

<option value="">12V</option>

</select>

</div>

<div class="input-group">

<span class="input-group-addon input-group-addon-control">面板</span>

<select id="bengkaiguan" onChange="bengfangxiang()" class="form-control input-sm">

<option value="chushui">20</option>

</select>

</div>

<div class="input-group">

<span class="input-group-addon input-group-addon-control">急停</span>

<select id="menzhuangtai" onChange="showdoor()" class="form-control input-sm input-sm">

<option value="guan">关</option>

</select>

</div>

<div class="input-group ">

<span class="input-group-addon input-group-addon-control ">供电</span>

<select id="bengkongzhi" onChange="begin()" class="form-control input-sm">

<option value="guan">正常</option>

</select>

</div>

<div class="input-group">

<span class="input-group-addon input-group-addon-control">运行</span>

<select class="form-control input-sm">

<option value="">正常</option>

</select>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="col-md-4" >

<table >

<tr>

<td>

<div class="panel panel-default panel-default-table">

<div class="panel-heading clearfix panel-heading-control">

<h3 class="panel-title panel-title-control">CTD</h3>

</div>

<div class="panel-body">

<table>

<tbody>

<tr>

<td>温度：</td>

<td>XXXX</td>

</tr>

<tr>

<td>湿度：</td>

<td>XXXX</td>

</tr>

<tr>

<td>风速</td>

<td>XXXX</td>

</tr>

<tr>

<td>风向：</td>

<td>XXXX</td>

</tr>

<tr>

<td>温度：</td>

<td>XXXX</td>

</tr>

<tr>

<td colspan="2"> 上次更新时间：</td>

</tr>

<tr>

<td colspan="2"> 2014-3-2 12:00:00</td>

</tr>

</tbody>

</table>

<div class="panel-body center-align-text">

<button type="button" class="btn btn-xs btn-default btn-default-control" data-original-title="" disabled>手动更新</button>

</div>

</div>

</div>

</td>

<td>

<div class="panel panel-default panel-default-table">

</div>

</td>

</tr>

</table>

</div>

</div>

</div>

</fieldset>

</section>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

<script language="javascript">

var dooropenfunction= function (){

if(document.getElementById("door\_left").width>2&&document.getElementById("menzhuangtai").options[document.getElementById("menzhuangtai").selectedIndex].value=="kai"){

document.getElementById("door\_left").width-=2;

document.getElementById("door\_right").width-=2;

}else{

dooropenfunction=null;

}

}

var doorclosfunction = function(){

if(document.getElementById("door\_left").width<26&&document.getElementById("menzhuangtai").options[document.getElementById("menzhuangtai").selectedIndex].value=="guan"){

document.getElementById("door\_left").width+=2;

document.getElementById("door\_right").width+=2;

}

else{

doorclosfunction=null;

}

}

function showdoor(){

if(document.getElementById("menzhuangtai").options[document.getElementById("menzhuangtai").selectedIndex].value=="kai"){

window.setInterval(dooropenfunction, 200);

}else if(document.getElementById("menzhuangtai").options[document.getElementById("menzhuangtai").selectedIndex].value=="guan"){

window.setInterval(doorclosfunction, 200);

}

}

</script>

<style>

.kongzhimianban-imgnew{

width:250px;

height:230px;

background-image:url(img/donghua1.jpg);

background-size: 250px 230px;

margin-left: 10px;

margin-bottom: -5px;

}

.kongzhimianban-img{

width:250px;

height:230px;

background-image:url(${ctx }/images/device/qianbiao.png);

background-size: 250px 230px;

margin-left: 10px;

margin-bottom: -5px;

}

.kongzhimianban-img2{

width:250px;

height:230px;

background-image:url(${ctx }/images/device/shexiangtou.png);

background-size: 250px 230px;

margin-left: 10px;

margin-bottom: -5px;

}

#door\_left{

margin-top: 66px;

margin-left: 146px;

}

#door\_right{

margin-top: 66px;

margin-left:-3px;

float: right;

margin-right: 20px;

}

#demo {

background: #FFF;

overflow:hidden;

border: 0;

width: 20px;

height:8px;

margin-left: 110px;

margin-top: -2px;

}

#indemo {

float: left;

width: 800%;

}

#demo1 {

float: left;

}

#demo2 {

float: left;

}

#demo1 img{

margin-top:-11px;

height:8px;

width:20px;

}

#demo2 img{

margin-top:-4px;

}

</style>

</html>