package com.sdocean.dataQuery.model;

import java.util.ArrayList;

import java.util.LinkedList;

import java.util.List;

public class StatisModel {

private String stationId;

private int statType;

private String beginDate;

private String endDate;

private List<StatData> datas; //饼形图数据

public List xtimes = new LinkedList(); //横坐标数据

public List<Object> ydatas = new ArrayList<Object>(); //纵坐标数据

private String indicatorIds;

public String getIndicatorIds() {

return indicatorIds;

}

public void setIndicatorIds(String indicatorIds) {

this.indicatorIds = indicatorIds;

}

public String getStationId() {

return stationId;

}

public void setStationId(String stationId) {

this.stationId = stationId;

}

public int getStatType() {

return statType;

}

public void setStatType(int statType) {

this.statType = statType;

}

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 List<StatData> getDatas() {

return datas;

}

public void setDatas(List<StatData> datas) {

this.datas = datas;

}

public List getXtimes() {

return xtimes;

}

public void setXtimes(List xtimes) {

this.xtimes = xtimes;

}

public List<Object> getYdatas() {

return ydatas;

}

public void setYdatas(List<Object> ydatas) {

this.ydatas = ydatas;

}

}

package com.sdocean.dataQuery.model;

public class Ydata {

private String value;

private String extra;

public String getValue() {

return value;

}

public void setValue(String value) {

this.value = value;

}

public String getExtra() {

return extra;

}

public void setExtra(String extra) {

this.extra = extra;

}

}

package com.sdocean.dataQuery.model;

public class Ydata {

private String value;

private String extra;

public String getValue() {

return value;

}

public void setValue(String value) {

this.value = value;

}

public String getExtra() {

return extra;

}

public void setExtra(String extra) {

this.extra = extra;

}

}

package com.sdocean.dataQuery.model;

public class Ydata {

private String value;

private String extra;

public String getValue() {

return value;

}

public void setValue(String value) {

this.value = value;

}

public String getExtra() {

return extra;

}

public void setExtra(String extra) {

this.extra = extra;

}

}

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;

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;

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;

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;

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.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.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.company.action;

import java.util.ArrayList;

import java.util.List;

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.company.model.CompanyModel;

import com.sdocean.company.model.SysCompanyModel;

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.page.model.PageResult;

import com.sdocean.page.model.UiColumn;

import com.sdocean.station.model.StationModel;

import com.sdocean.users.model.SysUser;

@Controller

public class CompanyAction {

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

@Autowired

private CompanyService companyService;

@Autowired

private OperationLogService logService;

@Autowired

private ConfigInfo info;

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

@ResponseBody

public String getCompanyList(HttpServletRequest request,

HttpServletResponse response){

List<SysCompanyModel> companys = companyService.getSysCompList();

return JsonUtil.toJson(companys);

}

/\*

\* 跳转到公司管理页面

\*/

@RequestMapping("info\_company.do")

public ModelAndView info\_company(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/company/companyInfo");

return mav;

}

/\*

\* 获得公司列表的查询结果

\*/

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

@ResponseBody

public String getgCompanyList(@ModelAttribute("model") CompanyModel model,HttpServletRequest request,

HttpServletResponse response){

PageResult result = new PageResult();

//添加表头

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

//添加数据

List<CompanyModel> list = companyService.getCompanyList(model);

result.setCols(cols);

result.setRows(list);

return JsonUtil.toJson(result);

}

/\*

\* 保存修改

\*/

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

@ResponseBody

public String saveCompanyChange(@ModelAttribute("model") CompanyModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = companyService.saveCompanyChange(model);

logService.saveOperationLog(result, request);

return result.getMessage();

}

/\*

\* 保存新增

\*/

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

@ResponseBody

public String saveNewCompany(@ModelAttribute("model") CompanyModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = companyService.saveNewCompany(model);

logService.saveOperationLog(result, request);

return result.getMessage();

}

/\*

\* 保存删除

\*/

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

@ResponseBody

public String deleCompany(@ModelAttribute("model") CompanyModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = companyService.deleCompany(model);

logService.saveOperationLog(result, request);

return result.getMessage();

}

/\*

\* 根据公司获得pcode树

\*/

@RequestMapping(value="showCompanyTree4Pcode.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String showCompanyTree4Pcode(@ModelAttribute("model") CompanyModel model,HttpServletRequest request,

HttpServletResponse response){

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

list = companyService.showCompanyTree4Pcode(model);

return JsonUtil.toJson(list);

}

/\*

\* 根据人员获得公司树

\*/

@RequestMapping(value="showComList4Users.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String showComList4Users(@ModelAttribute("model") SysUser model,HttpServletRequest request,

HttpServletResponse response){

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

list = companyService.showComList4Users(model);

return JsonUtil.toJson(list);

}

/\*

\* 根据站点获得公司树

\*/

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

@ResponseBody

public String geCompanyListByStation(@ModelAttribute("station") StationModel station,HttpServletRequest request,

HttpServletResponse response){

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

trees = companyService.geCompanyListByStation(station);

return JsonUtil.toJson(trees);

}

}

package com.sdocean.pollutant.model;

import java.util.List;

public class Pollutant4First {

private int wpId;

private String wpName;

private String collectDate;

private Double avgFlow; //一天的平均流速

private Double allFlow; //一天的水通量

private List<PollutantModel> polluList;

public Double getAvgFlow() {

return avgFlow;

}

public void setAvgFlow(Double avgFlow) {

this.avgFlow = avgFlow;

}

public Double getAllFlow() {

return allFlow;

}

public void setAllFlow(Double allFlow) {

this.allFlow = allFlow;

}

public List<PollutantModel> getPolluList() {

return polluList;

}

public void setPolluList(List<PollutantModel> polluList) {

this.polluList = polluList;

}

public int getWpId() {

return wpId;

}

public void setWpId(int wpId) {

this.wpId = wpId;

}

public String getWpName() {

return wpName;

}

public void setWpName(String wpName) {

this.wpName = wpName;

}

public String getCollectDate() {

return collectDate;

}

public void setCollectDate(String collectDate) {

this.collectDate = collectDate;

}

}

package com.sdocean.pollutant.model;

import java.util.List;

import com.sdocean.common.model.SelectTree;

import com.sdocean.device.model.DeviceModel;

public class PollutantModel {

private int id;

private String collectDate;

private int wpId;

private String wpName;

private int deviceId;

private String deviceName;

private String indicatorCode;

private String indicatorName;

private Double avgData;

private Double polluData;

private String beginDate;

private String endDate;

private List<SelectTree> indicatorTree;

private String idcicatorCods;

private int type;

private String typeName;

private String indicatorIds;

private List<DeviceModel> devices;

private String unit;

public String getTypeName() {

return typeName;

}

public void setTypeName(String typeName) {

this.typeName = typeName;

}

public String getUnit() {

return unit;

}

public void setUnit(String unit) {

this.unit = unit;

}

public String getIndicatorIds() {

return indicatorIds;

}

public void setIndicatorIds(String indicatorIds) {

this.indicatorIds = indicatorIds;

}

public List<DeviceModel> getDevices() {

return devices;

}

public void setDevices(List<DeviceModel> devices) {

this.devices = devices;

}

public int getType() {

return type;

}

public void setType(int type) {

this.type = type;

}

public String getIdcicatorCods() {

return idcicatorCods;

}

public void setIdcicatorCods(String idcicatorCods) {

this.idcicatorCods = idcicatorCods;

}

public List<SelectTree> getIndicatorTree() {

return indicatorTree;

}

public void setIndicatorTree(List<SelectTree> indicatorTree) {

this.indicatorTree = indicatorTree;

}

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 int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getCollectDate() {

return collectDate;

}

public void setCollectDate(String collectDate) {

this.collectDate = collectDate;

}

public int getWpId() {

return wpId;

}

public void setWpId(int wpId) {

this.wpId = wpId;

}

public String getWpName() {

return wpName;

}

public void setWpName(String wpName) {

this.wpName = wpName;

}

public int getDeviceId() {

return deviceId;

}

public void setDeviceId(int deviceId) {

this.deviceId = deviceId;

}

public String getDeviceName() {

return deviceName;

}

public void setDeviceName(String deviceName) {

this.deviceName = deviceName;

}

public String getIndicatorCode() {

return indicatorCode;

}

public void setIndicatorCode(String indicatorCode) {

this.indicatorCode = indicatorCode;

}

public String getIndicatorName() {

return indicatorName;

}

public void setIndicatorName(String indicatorName) {

this.indicatorName = indicatorName;

}

public Double getAvgData() {

return avgData;

}

public void setAvgData(Double avgData) {

this.avgData = avgData;

}

public Double getPolluData() {

return polluData;

}

public void setPolluData(Double polluData) {

this.polluData = polluData;

}

}

package com.sdocean.pollutant.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.device.model.DeviceModel;

import com.sdocean.page.model.UiColumn;

import com.sdocean.pollutant.dao.PollutantDao;

import com.sdocean.pollutant.model.Pollutant4First;

import com.sdocean.pollutant.model.PollutantModel;

import com.sdocean.station.model.StationModel;

@Service

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

public class PollutantService {

@Autowired

private PollutantDao pollutantDao;

/\*

\* 首页展示入海污染量

\*/

/\*

\* 为首页展示入海污染量提供数据

\*/

public Pollutant4First getPollutants4FirstPage(StationModel station){

Pollutant4First first = pollutantDao.getPollutants4FirstPage(station);

return first;

}

/\*

\* 入海污染源查询提供表头

\*/

public List<UiColumn> getColumns4Polluinfo(PollutantModel model){

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

cols = pollutantDao.getColumns4Polluinfo(model);

return cols;

}

/\*

\* 为入海污染量查询提供结果集

\*/

public List<Map<String, Object>> getRows4Polluinfo(PollutantModel model){

return pollutantDao.getRows4Polluinfo(model);

}

/\*

\* 为入海污染源设置查询提供表头

\*/

public List<UiColumn> getColumns4PolluSetinfo(PollutantModel model){

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

UiColumn col0 = new UiColumn("id", "id", false, "\*");

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

UiColumn col2 = new UiColumn("站点", "wpName", true, "\*");

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

UiColumn col6 = new UiColumn("设备", "deviceName", true, "\*");

UiColumn col7 = new UiColumn("indicatorCode", "indicatorCode", false, "\*");

UiColumn col8 = new UiColumn("参数", "indicatorName", true, "\*");

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

UiColumn col10 = new UiColumn("类型", "typeName", true, "\*");

cols.add(col0);

cols.add(col1);

cols.add(col2);

cols.add(col3);

cols.add(col6);

cols.add(col7);

cols.add(col8);

cols.add(col9);

cols.add(col10);

return cols;

}

/\*

\* 为入海污染源设置查询提供结果集

\*/

public List<PollutantModel> getPolluSetList(PollutantModel model){

return pollutantDao.getPolluSetList(model);

}

/\*

\* 添加入海污染源设置

\*/

public Result saveNewPolluSet(PollutantModel model){

return pollutantDao.saveNewPolluSet(model);

}

/\*

\* 修改入海污染源设置

\*/

public Result savePolluSet(PollutantModel model){

return pollutantDao.savePolluSet(model);

}

/\*

\* 删除入海污染源设置

\*/

public Result delePolluSet(PollutantModel model){

return pollutantDao.delePolluSet(model);

}

/\*

\* 为首页的入海污染物查询提供表头

\*/

public List<UiColumn> getColumns4Polluinfo4First(){

return pollutantDao.getColumns4Polluinfo4First();

}

/\*

\* 获得站点需要展示的入海污染量的信息

\*/

public List<DeviceModel> getDevices4Pollu(StationModel station){

return pollutantDao.getDevices4Pollu(station);

}

}

package com.sdocean.pollutant.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.device.model.DeviceModel;

import com.sdocean.page.model.UiColumn;

import com.sdocean.pollutant.dao.PollutantDao;

import com.sdocean.pollutant.model.Pollutant4First;

import com.sdocean.pollutant.model.PollutantModel;

import com.sdocean.station.model.StationModel;

@Service

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

public class PollutantService {

@Autowired

private PollutantDao pollutantDao;

/\*

\* 首页展示入海污染量

\*/

/\*

\* 为首页展示入海污染量提供数据

\*/

public Pollutant4First getPollutants4FirstPage(StationModel station){

Pollutant4First first = pollutantDao.getPollutants4FirstPage(station);

return first;

}

/\*

\* 入海污染源查询提供表头

\*/

public List<UiColumn> getColumns4Polluinfo(PollutantModel model){

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

cols = pollutantDao.getColumns4Polluinfo(model);

return cols;

}

/\*

\* 为入海污染量查询提供结果集

\*/

public List<Map<String, Object>> getRows4Polluinfo(PollutantModel model){

return pollutantDao.getRows4Polluinfo(model);

}

/\*

\* 为入海污染源设置查询提供表头

\*/

public List<UiColumn> getColumns4PolluSetinfo(PollutantModel model){

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

UiColumn col0 = new UiColumn("id", "id", false, "\*");

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

UiColumn col2 = new UiColumn("站点", "wpName", true, "\*");

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

UiColumn col6 = new UiColumn("设备", "deviceName", true, "\*");

UiColumn col7 = new UiColumn("indicatorCode", "indicatorCode", false, "\*");

UiColumn col8 = new UiColumn("参数", "indicatorName", true, "\*");

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

UiColumn col10 = new UiColumn("类型", "typeName", true, "\*");

cols.add(col0);

cols.add(col1);

cols.add(col2);

cols.add(col3);

cols.add(col6);

cols.add(col7);

cols.add(col8);

cols.add(col9);

cols.add(col10);

return cols;

}

/\*

\* 为入海污染源设置查询提供结果集

\*/

public List<PollutantModel> getPolluSetList(PollutantModel model){

return pollutantDao.getPolluSetList(model);

}

/\*

\* 添加入海污染源设置

\*/

public Result saveNewPolluSet(PollutantModel model){

return pollutantDao.saveNewPolluSet(model);

}

/\*

\* 修改入海污染源设置

\*/

public Result savePolluSet(PollutantModel model){

return pollutantDao.savePolluSet(model);

}

/\*

\* 删除入海污染源设置

\*/

public Result delePolluSet(PollutantModel model){

return pollutantDao.delePolluSet(model);

}

/\*

\* 为首页的入海污染物查询提供表头

\*/

public List<UiColumn> getColumns4Polluinfo4First(){

return pollutantDao.getColumns4Polluinfo4First();

}

/\*

\* 获得站点需要展示的入海污染量的信息

\*/

public List<DeviceModel> getDevices4Pollu(StationModel station){

return pollutantDao.getDevices4Pollu(station);

}

}

package com.sdocean.position.action;

import java.util.List;

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.RequestMapping;

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

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

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.position.model.SysPosition;

import com.sdocean.position.service.SysPositionService;

@Controller

public class SysPositionAction {

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

@Autowired

private SysPositionService positionService;

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

@ResponseBody

public String getPositionList(HttpServletRequest request,

HttpServletResponse response){

List<SysPosition> positions = positionService.getPositionList();

return JsonUtil.toJson(positions);

}

}

package com.sdocean.position.action;

import java.util.List;

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.RequestMapping;

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

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

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.position.model.SysPosition;

import com.sdocean.position.service.SysPositionService;

@Controller

public class SysPositionAction {

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

@Autowired

private SysPositionService positionService;

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

@ResponseBody

public String getPositionList(HttpServletRequest request,

HttpServletResponse response){

List<SysPosition> positions = positionService.getPositionList();

return JsonUtil.toJson(positions);

}

}

package com.sdocean.position.action;

import java.util.List;

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.RequestMapping;

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

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

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.position.model.SysPosition;

import com.sdocean.position.service.SysPositionService;

@Controller

public class SysPositionAction {

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

@Autowired

private SysPositionService positionService;

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

@ResponseBody

public String getPositionList(HttpServletRequest request,

HttpServletResponse response){

List<SysPosition> positions = positionService.getPositionList();

return JsonUtil.toJson(positions);

}

}

package com.sdocean.position.action;

import java.util.List;

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.RequestMapping;

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

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

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.position.model.SysPosition;

import com.sdocean.position.service.SysPositionService;

@Controller

public class SysPositionAction {

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

@Autowired

private SysPositionService positionService;

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

@ResponseBody

public String getPositionList(HttpServletRequest request,

HttpServletResponse response){

List<SysPosition> positions = positionService.getPositionList();

return JsonUtil.toJson(positions);

}

}

package com.sdocean.position.action;

import java.util.List;

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.RequestMapping;

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

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

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.position.model.SysPosition;

import com.sdocean.position.service.SysPositionService;

@Controller

public class SysPositionAction {

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

@Autowired

private SysPositionService positionService;

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

@ResponseBody

public String getPositionList(HttpServletRequest request,

HttpServletResponse response){

List<SysPosition> positions = positionService.getPositionList();

return JsonUtil.toJson(positions);

}

}

package com.sdocean.region.dao;

import java.util.ArrayList;

import java.util.List;

import jxl.biff.drawing.ComboBox;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.CombotreeModel;

import com.sdocean.common.model.SelectTree;

import com.sdocean.common.model.ZTreeModel;

import com.sdocean.dictionary.model.PublicModel;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.log.model.SysLoginLogModel;

import com.sdocean.menu.model.CurrMenu;

import com.sdocean.menu.model.SysMenu;

import com.sdocean.region.model.RegionModel;

import com.sdocean.role.model.RoleModel;

import com.sdocean.station.model.StationModel;

import com.sdocean.users.model.SysUser;

@Component

public class RegionDao extends OracleEngine{

/\*

\* 在下拉框中展示站点树

\*/

public List<SelectTree> getRegionList4Tree(StationModel station){

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

//得到第一层的地区代码

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.name as name,'true' as isExpanded,'true' as isActive,");

sql.append(" case when a.id =").append(station.getRegion\_id()).append(" then 'true' else 'false' end as selected ");

sql.append(" from g\_region a where level = 1");

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

for(SelectTree child:trees){

this.setChildren4Tree(child,station);

}

return trees;

}

/\*

\* 为当前的站点添加他的子元素

\*/

public void setChildren4Tree(SelectTree model,StationModel station){

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

StringBuffer sql = new StringBuffer("");

sql.append("select a.id,a.name as name,'true' as isExpanded,case when a.id = ").append(station.getRegion\_id()).append(" then 'true' else 'false' end as selected");

sql.append(" from g\_region a where a.pid = ").append(model.getId());

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

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

for(SelectTree children:child){

this.setChildren4Tree(children,station);

}

model.setChildren(child);

}

}

/\*

\* 查询ZTREE LIST中的上级元素

\*/

public List<ZTreeModel> getParentRegions(List<ZTreeModel> children){

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

//拼接PIDS

String pids ="0";

for(ZTreeModel child:children){

pids = pids + "," +child.getpId();

}

//查询当前等级的ZTREE

StringBuffer sql = new StringBuffer("");

sql.append(" select a.id,a.pid,'true' as open , a.name,'true' as nocheck");

sql.append(" from g\_region a");

sql.append(" where a.id in (").append(pids).append(")");

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

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

if(list.size()>0){

child = this.getParentRegions(list);

}

list.addAll(child);

return list;

}

/\*

\* 根据地区ID获得上级所有的地区信息

\*/

public List<RegionModel> getRegionListById(int id){

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

RegionModel region = new RegionModel();

StringBuffer sql = new StringBuffer();

sql.append(" select id,pid,code,name as text,level");

sql.append(" from g\_region ");

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

region = this.queryObject(sql.toString(), RegionModel.class);

if(region!=null){

list.add(region);

if(region.getPid()!=0){

list.addAll(this.getRegionListById(region.getPid()));

}

}

return list;

}

}

package com.sdocean.region.model;

public class RegionModel {

private int id;

private int pid;

private String code;

private String text;

private int level;

public RegionModel() {

super();

}

public RegionModel(int id, int pid, String code, String text, int level) {

super();

this.id = id;

this.pid = pid;

this.code = code;

this.text = text;

this.level = level;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public int getPid() {

return pid;

}

public void setPid(int pid) {

this.pid = pid;

}

public String getCode() {

return code;

}

public void setCode(String code) {

this.code = code;

}

public String getText() {

return text;

}

public void setText(String text) {

this.text = text;

}

public int getLevel() {

return level;

}

public void setLevel(int level) {

this.level = level;

}

@Override

public boolean equals(Object obj) {

RegionModel region = (RegionModel) obj;

if(this.id==region.id){

return true;

}else{

return false;

}

}

}

package com.sdocean.region.service;

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.SelectTree;

import com.sdocean.region.dao.RegionDao;

import com.sdocean.region.model.RegionModel;

import com.sdocean.station.model.StationModel;

@Service

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

public class RegionService {

@Autowired

private RegionDao regionDao;

/\*

\* 在下拉框中展示站点树

\*/

public List<SelectTree> getRegionList4Tree(StationModel station){

return regionDao.getRegionList4Tree(station);

}

/\*

\* 根据地区ID获得上级所有的地区信息

\*/

public List<RegionModel> getRegionListById(int id){

return regionDao.getRegionListById(id);

}

}

package com.sdocean.resources.dao;

import java.util.List;

import org.springframework.stereotype.Component;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.resources.dto.Resources;

@Component

public class ResourcesDao extends OracleEngine{

public List<Resources> findAll() {

String sqlString = "select r.id resid,r.name resname,r.url resurl,r.remark resremark,r.auth\_id authid,a.AUTHKEY from RESOURCES r left join AUTHORITY a on r.AUTH\_ID= a.id ";

List<Resources> queryObjectList = this.queryObjectList(sqlString, null, Resources.class);

return queryObjectList;

}

}

package com.sdocean.security;

import java.util.Collection;

import java.util.Iterator;

import org.springframework.security.access.AccessDecisionManager;

import org.springframework.security.access.AccessDeniedException;

import org.springframework.security.access.ConfigAttribute;

import org.springframework.security.authentication.InsufficientAuthenticationException;

import org.springframework.security.core.Authentication;

import org.springframework.security.core.GrantedAuthority;

public class MyAccessDecisionManager implements AccessDecisionManager {

public void decide(Authentication authentication, Object object, Collection<ConfigAttribute> configAttributes) throws AccessDeniedException, InsufficientAuthenticationException {

Authentication authentication1= authentication;

if(configAttributes == null) {

return;

}

//所请求的资源拥有的权限(这里一个资源只对应一个权限)

Iterator<ConfigAttribute> iterator = configAttributes.iterator();

while(iterator.hasNext()) {

ConfigAttribute configAttribute = iterator.next();

//访问所请求资源所需要的权限

String needPermission = configAttribute.getAttribute();

//用户所拥有的权限authentication

for(GrantedAuthority ga : authentication.getAuthorities()) {

if(needPermission.equals(ga.getAuthority())) {

return;

}

}

}

//没有权限

throw new AccessDeniedException(" 没有权限访问！ ");

}

public boolean supports(ConfigAttribute attribute) {

// TODO Auto-generated method stub

return true;

}

public boolean supports(Class<?> clazz) {

// TODO Auto-generated method stub

return true;

}

}

package com.sdocean.security;

import java.io.IOException;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import org.springframework.security.access.SecurityMetadataSource;

import org.springframework.security.access.intercept.AbstractSecurityInterceptor;

import org.springframework.security.access.intercept.InterceptorStatusToken;

import org.springframework.security.web.FilterInvocation;

import org.springframework.security.web.access.intercept.FilterInvocationSecurityMetadataSource;

public class MySecurityFilter extends AbstractSecurityInterceptor implements Filter {

//与applicationContext-security.xml里的myFilter的属性securityMetadataSource对应，

//其他的两个组件，已经在AbstractSecurityInterceptor定义

private FilterInvocationSecurityMetadataSource securityMetadataSource;

@Override

public SecurityMetadataSource obtainSecurityMetadataSource() {

return this.securityMetadataSource;

}

public void doFilter(ServletRequest request, ServletResponse response,

FilterChain chain) throws IOException, ServletException {

FilterInvocation fi = new FilterInvocation(request, response, chain);

invoke(fi);

}

private void invoke(FilterInvocation fi) throws IOException, ServletException {

// object为FilterInvocation对象

//super.beforeInvocation(fi);源码

//1.获取请求资源的权限

//执行Collection<ConfigAttribute> attributes = SecurityMetadataSource.getAttributes(object);

//2.是否拥有权限

//this.accessDecisionManager.decide(authenticated, object, attributes);

InterceptorStatusToken token = super.beforeInvocation(fi);

try {

fi.getChain().doFilter(fi.getRequest(), fi.getResponse());

} finally {

super.afterInvocation(token, null);

}

}

public FilterInvocationSecurityMetadataSource getSecurityMetadataSource() {

return securityMetadataSource;

}

public void setSecurityMetadataSource(FilterInvocationSecurityMetadataSource securityMetadataSource) {

this.securityMetadataSource = securityMetadataSource;

}

public void init(FilterConfig arg0) throws ServletException {

// TODO Auto-generated method stub

}

public void destroy() {

// TODO Auto-generated method stub

}

@Override

public Class<? extends Object> getSecureObjectClass() {

//下面的MyAccessDecisionManager的supports方面必须放回true,否则会提醒类型错误

return FilterInvocation.class;

}

}

package com.sdocean.security;

import java.util.ArrayList;

import java.util.Collection;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import org.springframework.security.access.ConfigAttribute;

import org.springframework.security.access.SecurityConfig;

import org.springframework.security.web.FilterInvocation;

import org.springframework.security.web.access.intercept.FilterInvocationSecurityMetadataSource;

import com.sdocean.resources.dto.Resources;

import com.sdocean.resources.service.ResourcesManager;

//1 加载资源与权限的对应关系

public class MySecurityMetadataSource implements FilterInvocationSecurityMetadataSource {

//由spring调用

public MySecurityMetadataSource(ResourcesManager resourcesManager) {

this.resourcesManager = resourcesManager;

loadResourceDefine();

}

private ResourcesManager resourcesManager;

private static Map<String, Collection<ConfigAttribute>> resourceMap = null;

public void setResourcesDao(ResourcesManager resourcesManager) {

this.resourcesManager = resourcesManager;

}

public Collection<ConfigAttribute> getAllConfigAttributes() {

// TODO Auto-generated method stub

return null;

}

public boolean supports(Class<?> clazz) {

// TODO Auto-generated method stub

return true;

}

//加载所有资源与权限的关系

private void loadResourceDefine() {

if(resourceMap == null) {

resourceMap = new HashMap<String, Collection<ConfigAttribute>>();

List<Resources> resources = this.resourcesManager.findAll();

for (Resources resource : resources) {

Collection<ConfigAttribute> configAttributes = new ArrayList<ConfigAttribute>();

//以权限名封装为Spring的security Object

ConfigAttribute configAttribute = new SecurityConfig(resource.getAUTHKEY());

configAttributes.add(configAttribute);

resourceMap.put(resource.getResUrl(), configAttributes);

}

}

// Set<Entry<String, Collection<ConfigAttribute>>> resourceSet = resourceMap.entrySet();

// Iterator<Entry<String, Collection<ConfigAttribute>>> iterator = resourceSet.iterator();

}

//返回所请求资源所需要的权限

public Collection<ConfigAttribute> getAttributes(Object object) throws IllegalArgumentException {

String requestUrl = ((FilterInvocation) object).getRequestUrl();

if(resourceMap == null) {

loadResourceDefine();

}

return resourceMap.get(requestUrl);

}

}

package com.sdocean.security;

import java.util.Collection;

import java.util.HashSet;

import java.util.Set;

import org.springframework.security.core.GrantedAuthority;

import org.springframework.security.core.userdetails.User;

import org.springframework.security.core.userdetails.UserDetails;

import org.springframework.security.core.userdetails.UserDetailsService;

import org.springframework.security.core.userdetails.UsernameNotFoundException;

import com.sdocean.users.model.SysUser;

import com.sdocean.users.service.UsersManager;

public class MyUserDetailServiceImpl implements UserDetailsService {

private UsersManager usersManager;

public void setUsersManager(UsersManager usersManager) {

this.usersManager = usersManager;

}

public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {

SysUser users = this.usersManager.getUsersByAccount(username);

if(users == null) {

throw new UsernameNotFoundException(username);

}

Collection<GrantedAuthority> grantedAuths = obtionGrantedAuthorities(users);

boolean enables = true;

boolean accountNonExpired = true;

boolean credentialsNonExpired = true;

boolean accountNonLocked = true;

User userdetail = new User(users.getUserName(), users.getPassword(), enables, accountNonExpired, credentialsNonExpired, accountNonLocked, grantedAuths);

return userdetail;

}

//取得用户的权限

private Set<GrantedAuthority> obtionGrantedAuthorities(SysUser user) {

Set<GrantedAuthority> authSet = new HashSet<GrantedAuthority>();

/\*List<Authority> authoritysByUserAccount = this.authorityManager.getAuthoritysByUserAccount(user);

for (Authority authority : authoritysByUserAccount) {

authSet.add(new SimpleGrantedAuthority(authority.getAuthKey()));

}\*/

return authSet;

}

}

package com.sdocean.security;

import java.io.IOException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.apache.commons.lang.StringUtils;

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

import org.springframework.security.authentication.AuthenticationServiceException;

import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;

import org.springframework.security.core.Authentication;

import org.springframework.security.core.AuthenticationException;

import org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.log.service.SysLoginLogService;

import com.sdocean.users.model.SysUser;

import com.sdocean.users.service.UsersManager;

public class MyUsernamePasswordAuthenticationFilter extends UsernamePasswordAuthenticationFilter{

public static final String VALIDATE\_CODE = "validateCode";

public static final String USERNAME = "username";

public static final String PASSWORD = "password";

private UsersManager usersManager;

public void setUsersManager(UsersManager usersManager) {

this.usersManager = usersManager;

}

private SysLoginLogService loginLogService;

public void setLoginLogService(SysLoginLogService loginLogService){

this.loginLogService = loginLogService;

}

@Override

public Authentication attemptAuthentication(HttpServletRequest request, HttpServletResponse response) throws AuthenticationException {

if (!request.getMethod().equals("POST")) {

throw new AuthenticationServiceException("Authentication method not supported: " + request.getMethod());

}

//检测验证码

// checkValidateCode(request);

String username = obtainUsername(request);

String password = obtainPassword(request);

//验证用户账号与密码是否对应

username = username.trim();

SysUser users = this.usersManager.getUsersByAccount(username);

if(users == null || !users.getPassword().equals(password)) {

throw new AuthenticationServiceException("用户名或者密码错误！");

}

//将用户信息存入到session中

HttpSession session = request.getSession();

session.setAttribute("user", users);

UsernamePasswordAuthenticationToken authRequest = new UsernamePasswordAuthenticationToken(username, password);

// 允许子类设置详细属性

setDetails(request, authRequest);

//loginLogService.saveSysLoginLog(request);

return this.getAuthenticationManager().authenticate(authRequest);

}

protected void checkValidateCode(HttpServletRequest request) {

HttpSession session = request.getSession();

String sessionValidateCode = obtainSessionValidateCode(session);

//让上一次的验证码失效

session.setAttribute(VALIDATE\_CODE, null);

String validateCodeParameter = obtainValidateCodeParameter(request);

if (StringUtils.isEmpty(validateCodeParameter) || !sessionValidateCode.equalsIgnoreCase(validateCodeParameter)) {

throw new AuthenticationServiceException("验证码错误！");

}

}

private String obtainValidateCodeParameter(HttpServletRequest request) {

Object obj = request.getParameter(VALIDATE\_CODE);

return null == obj ? "" : obj.toString();

}

protected String obtainSessionValidateCode(HttpSession session) {

Object obj = session.getAttribute(VALIDATE\_CODE);

return null == obj ? "" : obj.toString();

}

@Override

protected String obtainUsername(HttpServletRequest request) {

Object obj = request.getParameter(USERNAME);

return null == obj ? "" : obj.toString();

}

@Override

protected String obtainPassword(HttpServletRequest request) {

Object obj = request.getParameter(PASSWORD);

return null == obj ? "" : obj.toString();

}

}

package com.sdocean.security;

import java.io.IOException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import javax.servlet.http.HttpSession;

import org.apache.commons.lang.StringUtils;

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

import org.springframework.security.authentication.AuthenticationServiceException;

import org.springframework.security.authentication.UsernamePasswordAuthenticationToken;

import org.springframework.security.core.Authentication;

import org.springframework.security.core.AuthenticationException;

import org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter;

import com.sdocean.frame.model.ConfigInfo;

import com.sdocean.log.service.SysLoginLogService;

import com.sdocean.users.model.SysUser;

import com.sdocean.users.service.UsersManager;

public class MyUsernamePasswordAuthenticationFilter extends UsernamePasswordAuthenticationFilter{

public static final String VALIDATE\_CODE = "validateCode";

public static final String USERNAME = "username";

public static final String PASSWORD = "password";

private UsersManager usersManager;

public void setUsersManager(UsersManager usersManager) {

this.usersManager = usersManager;

}

private SysLoginLogService loginLogService;

public void setLoginLogService(SysLoginLogService loginLogService){

this.loginLogService = loginLogService;

}

@Override

public Authentication attemptAuthentication(HttpServletRequest request, HttpServletResponse response) throws AuthenticationException {

if (!request.getMethod().equals("POST")) {

throw new AuthenticationServiceException("Authentication method not supported: " + request.getMethod());

}

//检测验证码

// checkValidateCode(request);

String username = obtainUsername(request);

String password = obtainPassword(request);

//验证用户账号与密码是否对应

username = username.trim();

SysUser users = this.usersManager.getUsersByAccount(username);

if(users == null || !users.getPassword().equals(password)) {

throw new AuthenticationServiceException("用户名或者密码错误！");

}

//将用户信息存入到session中

HttpSession session = request.getSession();

session.setAttribute("user", users);

UsernamePasswordAuthenticationToken authRequest = new UsernamePasswordAuthenticationToken(username, password);

// 允许子类设置详细属性

setDetails(request, authRequest);

//loginLogService.saveSysLoginLog(request);

return this.getAuthenticationManager().authenticate(authRequest);

}

protected void checkValidateCode(HttpServletRequest request) {

HttpSession session = request.getSession();

String sessionValidateCode = obtainSessionValidateCode(session);

//让上一次的验证码失效

session.setAttribute(VALIDATE\_CODE, null);

String validateCodeParameter = obtainValidateCodeParameter(request);

if (StringUtils.isEmpty(validateCodeParameter) || !sessionValidateCode.equalsIgnoreCase(validateCodeParameter)) {

throw new AuthenticationServiceException("验证码错误！");

}

}

private String obtainValidateCodeParameter(HttpServletRequest request) {

Object obj = request.getParameter(VALIDATE\_CODE);

return null == obj ? "" : obj.toString();

}

protected String obtainSessionValidateCode(HttpSession session) {

Object obj = session.getAttribute(VALIDATE\_CODE);

return null == obj ? "" : obj.toString();

}

@Override

protected String obtainUsername(HttpServletRequest request) {

Object obj = request.getParameter(USERNAME);

return null == obj ? "" : obj.toString();

}

@Override

protected String obtainPassword(HttpServletRequest request) {

Object obj = request.getParameter(PASSWORD);

return null == obj ? "" : obj.toString();

}

}

package com.sdocean.sms.action;

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.menu.model.CurrMenu;

import com.sdocean.page.model.PageResult;

import com.sdocean.page.model.UiColumn;

import com.sdocean.sms.model.SmsLog;

import com.sdocean.sms.model.SmsMouldModel;

import com.sdocean.sms.model.SmsSettingModel;

import com.sdocean.sms.service.SmsService;

import com.sdocean.users.model.SysUser;

@Controller

public class SmsAction {

@Resource

SmsService smsService;

@Resource

OperationLogService logService;

@Autowired

private ConfigInfo info;

@RequestMapping("info\_smssetting.do")

public ModelAndView info\_smssetting(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/sms/smsSettingInfo");

return mav;

}

@RequestMapping("info\_smssetting4user.do")

public ModelAndView info\_smssetting4user(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getPageVision()+"/sms/smsSetting4userInfo");

CurrMenu menu = new CurrMenu();

menu.setCurl("info\_smssetting4user.do");

HttpSession session = request.getSession();

session.setAttribute("currMenu", menu);

return mav;

}

/\*

\* 跳转到短信发送记录页面

\*/

@RequestMapping("info\_smslog.do")

public ModelAndView info\_smslog(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/sms/smsLogInfo");

return mav;

}

/\*

\* 展示短信配置中,某站点的人员列表

\*/

@RequestMapping(value="showUsers4SmsSetting.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String showUsers4SmsSetting(@ModelAttribute("model") SmsSettingModel model,HttpServletRequest request,

HttpServletResponse response){

PageResult page = new PageResult();

//获得当前的人员信息

HttpSession session = request.getSession();

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

//得到展示表格的表头

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

//得到展示表格的结果集

List<SmsSettingModel> rows = smsService.getUsers4SmsSetting(model, user);

page.setCols(cols);

page.setRows(rows);

return JsonUtil.toJson(page);

}

/\*

\* 初始化当前用户的短信配置

\*/

@RequestMapping(value="getSmsSetting4User.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String getSmsSetting4User(HttpServletRequest request,

HttpServletResponse response){

HttpSession session = request.getSession();

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

SmsSettingModel setting = new SmsSettingModel();

setting = smsService.infoUserSetting(model);

return JsonUtil.toJson(setting);

}

/\*

\* 初始化用户下的短信配置

\*/

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

@ResponseBody

public String infoUserSetting(@ModelAttribute("model") SysUser model,HttpServletRequest request,

HttpServletResponse response){

SmsSettingModel setting = new SmsSettingModel();

setting = smsService.infoUserSetting(model);

return JsonUtil.toJson(setting);

}

/\*

\* 保存用户短信配置

\*/

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

@ResponseBody

public String saveUserSetting(@ModelAttribute("model") SmsSettingModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = smsService.saveUserSetting(model);

logService.saveOperationLog(result, request);

return result.getMessage();

}

/\*

\* 跳转到短信模板设置页面

\*/

@RequestMapping("info\_smsmould.do")

public ModelAndView info\_smsmould(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/sms/smsMouldInfo");

return mav;

}

/\*

\* 展示模板列表

\*/

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

@ResponseBody

public String showSmsMouldList(@ModelAttribute("model") SmsMouldModel model,HttpServletRequest request,

HttpServletResponse response){

PageResult page = new PageResult();

//得到展示表格的表头

List<UiColumn> cols = smsService.getCols4MouldList();

//得到展示表格的结果集

List<SmsMouldModel> rows = smsService.getSmsMoulds4Rows(model);

page.setCols(cols);

page.setRows(rows);

return JsonUtil.toJson(page);

}

/\*

\* 保存短信模板修改

\*/

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

@ResponseBody

public String saveSmsMouldChange(@ModelAttribute("model") SmsMouldModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = smsService.saveSmsMouldChange(model);

logService.saveOperationLog(result, request);

return result.getMessage();

}

/\*

\* 保存短信模板新增

\*/

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

@ResponseBody

public String saveNewSmsMould(@ModelAttribute("model") SmsMouldModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = smsService.saveNewSmsMould(model);

logService.saveOperationLog(result, request);

return result.getMessage();

}

@RequestMapping(value="saveSmsSetting4User.do", method = RequestMethod.POST,produces = "application/json;charset=UTF-8")

@ResponseBody

public String saveSmsSetting4User(@ModelAttribute("model") SmsSettingModel model,HttpServletRequest request,

HttpServletResponse response){

HttpSession session = request.getSession();

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

model.setUserId(user.getId());

Result result = new Result();

result = smsService.saveUserSetting(model);

return result.getMessage();

}

/\*

\* 得到短信发送记录列表

\*/

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

@ResponseBody

public String getSmsLogs(@ModelAttribute("model") SmsLog model,HttpServletRequest request,

HttpServletResponse response){

PageResult page = new PageResult();

List<UiColumn> cols = smsService.getCols4LogList();

List<SmsLog> rows = smsService.getSmsLogs(model);

page.setCols(cols);

page.setRows(rows);

return JsonUtil.toJson(page);

}

/\*

\* 跳转到短信发送配置

\*/

@RequestMapping("info\_smsstation.do")

public ModelAndView info\_smsstation(HttpServletRequest request,

HttpServletResponse response)throws Exception{

ModelAndView mav = new ModelAndView("/"+info.getManageVision()+"/sms/smsStationInfo");

return mav;

}

/\*

\* 保存短信发送配置

\*/

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

@ResponseBody

public String saveSmsStationSetting(@ModelAttribute("model") SmsSettingModel model,HttpServletRequest request,

HttpServletResponse response){

Result result = new Result();

result = smsService.saveSmsStationSetting(model);

logService.saveOperationLog(result, request);

return JsonUtil.toJson(result);

}

}

package com.sdocean.sms.dao;

import java.util.ArrayList;

import java.util.List;

import org.springframework.stereotype.Component;

import com.sdocean.common.model.Result;

import com.sdocean.frame.dao.OracleEngine;

import com.sdocean.frame.util.JsonUtil;

import com.sdocean.role.model.RoleModel;

import com.sdocean.sms.model.SmsLog;

import com.sdocean.sms.model.SmsMouldModel;

import com.sdocean.sms.model.SmsSettingModel;

import com.sdocean.station.model.StationModel;

import com.sdocean.users.model.SysUser;

import freemarker.template.utility.StringUtil;

@Component

public class SmsDao extends OracleEngine{

/\*

\* 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){

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

StringBuffer sql = new StringBuffer("");

sql.append(" select a.userid,' 默认' as userName,'' as telephone,");

sql.append(" a.betweentime,a.ambegin,a.amend,a.pmbegin,a.pmend,");

sql.append(" a.mon,a.tues,a.wed,a.thur,a.fri,a.satur,a.sun,");

sql.append(" case a.isactive when 1 then '接收' else '不接收' end as isactiveName");

sql.append(" from sys\_sms\_userrole a");

sql.append(" where a.userid = 0");

sql.append(" union all");

sql.append(" select a.userid,b.realname as userName,b.telephone,");

sql.append(" a.betweentime,a.ambegin,a.amend,a.pmbegin,a.pmend,");

sql.append(" a.mon,a.tues,a.wed,a.thur,a.fri,a.satur,a.sun,");

sql.append(" case a.isactive when 1 then '接收' else '不接收' end as isactiveName");

sql.append(" from sys\_sms\_userrole a,sys\_user b");

sql.append(" where a.userid = b.id and b.isactive = 1");

//添加查询条件

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

sql.append(" and (b.username like '%").append(model.getUserName()).append("%'");

sql.append(" or b.realname like '%").append(model.getUserName()).append("%')");

}

//添加排序

sql.append(" order by userName");

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

return list;

}

/\*

\* 初始化用户下的短信配置

\*/

public SmsSettingModel infoUserSetting(SysUser user){

SmsSettingModel result = new SmsSettingModel();

//读取发送规则配置

StringBuffer rolesql = new StringBuffer("");

rolesql.append(" select userid,ambegin,amend,pmbegin,pmend,mon,tues,wed,thur,fri,satur,sun,betweentime,isactive from sys\_sms\_userrole where userid =").append(user.getId());

result = this.queryObject(rolesql.toString(), SmsSettingModel.class);

if(result==null){

StringBuffer role0sql = new StringBuffer("");

role0sql.append(" select userid,ambegin,amend,pmbegin,pmend,mon,tues,wed,thur,fri,satur,sun,betweentime,isactive from sys\_sms\_userrole where userid = 0");

result = this.queryObject(role0sql.toString(), SmsSettingModel.class);

}

return result;

}

/\*

\* 保存用户短信配置

\*/

public Result saveUserSetting(SmsSettingModel model){

//对返回结果进行初始化

Result result = new Result();

result.setDotype(result.UPDATE);

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

result.setResult(result.SUCCESS);

result.setMessage("保存成功");

//保存配置信息

//保存发送时间规则设置

StringBuffer roleSql = new StringBuffer("");

roleSql.append(" insert into sys\_sms\_userrole(userid,ambegin,amend,pmbegin,pmend,mon,tues,wed,thur,fri,satur,sun,betweentime,isactive)");

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

roleSql.append(" on duplicate key update ambegin=values(ambegin),amend=values(amend),pmbegin=values(pmbegin),");

roleSql.append(" pmend=values(pmend),mon=values(mon),tues=values(tues),wed=values(wed),thur=values(thur),fri=values(fri),satur=values(satur),sun=values(sun),");

roleSql.append(" betweentime=values(betweentime),isactive=values(isactive)");

int roleres = 0;

Object[] roleParams = new Object[]{

model.getUserId(),model.getAmbegin(),model.getAmend(),model.getPmbegin(),

model.getPmend(),model.getMon(),model.getTues(),model.getWed(),model.getThur(),

model.getFri(),model.getSatur(),model.getSun(),model.getBetweenTime(),model.getIsactive()

};

try {

roleres = this.update(roleSql.toString(), roleParams);

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("保存失败");

return result;

}

return result;

}

/\*

\* 查询符合条件的短信模板列表

\*/

public List<SmsMouldModel> getSmsMoulds4Rows(SmsMouldModel model){

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

if(model!=null&&model.getStationId()<10000){

//首先添加默认站点stationid = 0

StringBuffer first = new StringBuffer("");

first.append(" select a.id,a.stationid,'默认站点' as stationName,a.type,c.value as typename,a.mould");

first.append(" from sys\_sms\_mould a,sys\_public c");

first.append(" where c.parentcode = '0010' and a.type = c.classid");

first.append(" and a.stationid = 0");

//添加查询条件

if(model.getType()>0){

first.append(" and a.type =").append(model.getType());

}

list = this.queryObjectList(first.toString(), SmsMouldModel.class);

}

//查询非默认站点的列表

List<SmsMouldModel> nlist = new ArrayList<SmsMouldModel>();

StringBuffer sql = new StringBuffer();

sql.append(" select a.id,a.stationid,b.title as stationName,a.type,c.value as typename,a.mould");

sql.append(" from sys\_sms\_mould a ,aiot\_watch\_point b,sys\_public c");

sql.append(" where a.stationid = b.id and b.isactive = 1 and c.parentcode = '0010' and a.type = c.classid");

//添加查询条件

if(model!=null&&model.getStationId()>10000){

sql.append(" and a.stationid = ").append(model.getStationId());

}

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

sql.append(" and a.type = ").append(model.getType());

}

nlist = this.queryObjectList(sql.toString(), SmsMouldModel.class);

list.addAll(nlist);

return list;

}

/\*

\* 对短信发送模板修改

\*/

public Result saveSmsMouldChange(SmsMouldModel 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");

//验证station以及type唯一性

StringBuffer checksql = new StringBuffer("");

checksql.append(" select count(1) from sys\_sms\_mould where stationid =").append(model.getStationId());

checksql.append(" and type = ").append(model.getType()).append(" and id <>").append(model.getId());

int chect =0;

try {

chect = this.queryForInt(checksql.toString(), null);

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("站点类型唯一性检查时失败");

return result;

}

if(chect>0){

result.setResult(result.FAILED);

result.setMessage("违反站点,类型唯一性原则");

return result;

}

//提交数据

StringBuffer sql = new StringBuffer("");

sql.append(" update sys\_sms\_mould set stationid=?,type=?,mould=? where id=?");

Object[] params = new Object[]{

model.getStationId(),model.getType(),model.getMould(),model.getId()

};

int res = 0;

try {

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

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("修改失败");

}

return result;

}

/\*

\* 保存新增的短信发送模板

\*/

public Result saveNewSmsMould(SmsMouldModel 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");

//验证station以及type唯一性

StringBuffer checksql = new StringBuffer("");

checksql.append(" select count(1) from sys\_sms\_mould where stationid =").append(model.getStationId());

checksql.append(" and type = ").append(model.getType());

int chect =0;

try {

chect = this.queryForInt(checksql.toString(), null);

} catch (Exception e) {

result.setResult(result.FAILED);

result.setMessage("站点类型唯一性检查时失败");

return result;

}

if(chect>0){

result.setResult(result.FAILED);

result.setMessage("违反站点,类型唯一性原则");

return result;

}

//提交数据

StringBuffer sql = new StringBuffer("");

sql.append(" insert into sys\_sms\_mould(stationid,type,mould) values(?,?,?)");

sql.append(" on duplicate key update stationId=values(stationid),type=values(type),mould=values(mould)");

Object[] params = new Object[]{

model.getStationId(),model.getType(),model.getMould()

};

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;

}

/\*

\* 查询短信发送记录,

\* 增加模糊查询

\*/

public List<SmsLog> getSmsLogs(SmsLog model){

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

StringBuffer sql = new StringBuffer("");

sql.append(" select a.userid,b.realname as username,a.telephone,a.title,a.message,a.sendtime,");

sql.append(" a.createtime,a.ifsend,case a.ifsend when 0 then '未发送' else '发送' end as ifSendName,a.error");

sql.append(" from sys\_sms a,sys\_user b");

sql.append(" where a.userid = b.id");

//增加模糊查询条件

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

sql.append(" and (b.realname like '%").append(model.getUserName()).append("%' ");

sql.append(" or a.telephone like '%").append(model.getUserName()).append("%' )");

}

//增加排序

sql.append(" order by createtime desc ");

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

return list;

}

/\*

\* 保存短信发送配置

\*/

public Result saveSmsStationSetting(SmsSettingModel 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("保存成功");

//删除该站点,该类型的相关短信接收配置

StringBuffer dsql = new StringBuffer("");

dsql.append(" delete from sys\_sms\_station\_user ");

dsql.append(" where stationid = ").append(model.getStationId());

dsql.append(" and type = ").append(model.getType());

int dres = 0;

try {

dres = this.update(dsql.toString(), null);

} catch (Exception e) {

e.printStackTrace();

result.setResult(Result.FAILED);

result.setMessage("删除原有数据时失败");

return result;

}

//添加短信接收配置

StringBuffer asql = new StringBuffer("");

asql.append(" insert into sys\_sms\_station\_user(stationid,userid,type)");

asql.append(" values(0,0,0)");

String userIds = model.getUserIds();

String[] users = userIds.split(",");

for(int i=0;i<users.length;i++){

asql.append(",(").append(model.getStationId()).append(",").append(users[i]).append(",").append(model.getType()).append(")");

}

asql.append(" on duplicate key update userid = values(userid)");

int ares = 0;

try {

ares = this.update(asql.toString(), null);

} catch (Exception e) {

e.printStackTrace();

result.setResult(Result.FAILED);

result.setMessage("添加短信接收配置时失败");

}

return result;

}

}

package com.sdocean.sms.model;

public class SmsLog {

private int id;

private int userId;

private String userName;

private String telephone;

private String title;

private String message;

private String sendTime;

private String createTime;

private int ifSend;

private String ifSendName;

private String error;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

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 String getTelephone() {

return telephone;

}

public void setTelephone(String telephone) {

this.telephone = telephone;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getMessage() {

return message;

}

public void setMessage(String message) {

this.message = message;

}

public String getSendTime() {

return sendTime;

}

public void setSendTime(String sendTime) {

this.sendTime = sendTime;

}

public String getCreateTime() {

return createTime;

}

public void setCreateTime(String createTime) {

this.createTime = createTime;

}

public int getIfSend() {

return ifSend;

}

public void setIfSend(int ifSend) {

this.ifSend = ifSend;

}

public String getIfSendName() {

return ifSendName;

}

public void setIfSendName(String ifSendName) {

this.ifSendName = ifSendName;

}

public String getError() {

return error;

}

public void setError(String error) {

this.error = error;

}

}

package com.sdocean.sms.model;

public class SmsMouldModel {

private int id;

private int stationId;

private String stationName;

private int type;

private String typeName;

private String mould;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

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 getType() {

return type;

}

public void setType(int type) {

this.type = type;

}

public String getTypeName() {

return typeName;

}

public void setTypeName(String typeName) {

this.typeName = typeName;

}

public String getMould() {

return mould;

}

public void setMould(String mould) {

this.mould = mould;

}

}

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>