package com.yawa.busi.controller;

import java.util.List;

import javax.servlet.http.HttpServletRequest;

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.ResponseBody;

import com.yawa.busi.model.CategoriesVO;

import com.yawa.busi.service.CategoryService;

import com.yawa.core.model.User;

@Controller

@RequestMapping("/category")

public class CategoryController {

@Autowired

private CategoryService categoryService;

@RequestMapping(value = "/getCategoryList")

@ResponseBody

public List<CategoriesVO> getCategoryList(HttpServletRequest request) {

User user = (User) request.getSession().getAttribute("currentUser");

CategoriesVO c = new CategoriesVO();

c.setVendor\_id(Integer.parseInt(user.getVendor\_id()));

List<CategoriesVO> list = this.categoryService.getList(c);

return list;

}

}

package com.yawa.busi.controller;

import java.util.HashMap;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

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.ResponseBody;

import com.yawa.busi.model.Customer;

import com.yawa.busi.service.CustomerService;

import com.yawa.core.model.User;

import com.yawa.util.model.Page;

/\*\*

\* 客户 ->客户的查询以及新增

\* @author am

\* @since 2014-09-11

\* @desc 客户查询列表 、客户的新增

\*/

@Controller

@RequestMapping("/customer")

public class CustomerController {

@Autowired

private CustomerService customerService;

@RequestMapping("/queryList")

@ResponseBody

public Page<Customer> getList(HttpServletRequest request,int start,int limit,String name,int cummunityid){

User user = (User) request.getSession().getAttribute("currentUser");

Map<String,Object> params = new HashMap<String, Object>();

params.put("customer\_name", name);

params.put("community\_id", cummunityid);

params.put("vendor\_id",Integer.parseInt(user.getVendor\_id()));

return customerService.getPageList(params, start, limit);

}

//回显

@RequestMapping("/getById")

@ResponseBody

public Customer getById(Integer id){

return this.customerService.findEntity(id);

}

/\*\*

\* 新增一条记录

\* @param request

\* @param govinfoVO

\* @return

\*/

@RequestMapping(value = "/insert")

@ResponseBody

public String updateContractInfo(String name,String phone,String address,String memo,Integer community,String categories) {

String flag = "true";

Customer c=new Customer();

c.setCustomer\_name(name);

c.setMobile(phone);

c.setMemo(memo);

c.setAddress(address);

c.setTag(categories);

c.setCommunity\_id(community);

this.customerService.insertEntity(c);

return flag;

}

}

package com.yawa.busi.controller;

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

import org.springframework.stereotype.Controller;

import org.springframework.util.StringUtils;

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

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

import com.yawa.core.model.User;

import com.yawa.core.service.UserService;

import com.yawa.http.model.CommExchange;

import com.yawa.http.service.ExchangeService;

import com.yawa.util.BeanUtils;

import com.yawa.util.context.SecurityContextUtil;

import com.yawa.util.model.Page;

import com.yawa.util.model.ResponseData;

@Controller

@RequestMapping("/exchange")

public class ExchangeController {

@Autowired

private ExchangeService exchangeService;

@Autowired

private UserService userService;

/\*\*

\* 分页查询列表

\* @param start

\* @param limit

\* @param exchange

\* @return

\*/

@RequestMapping("/getPageList")

@ResponseBody

public Page<CommExchange> getPageList(int start,int limit,CommExchange exchange){

Integer user\_id = SecurityContextUtil.getCurrentUser().getId();

User logUser = this.userService.findEntity(user\_id);

if(StringUtils.isEmpty(logUser.getVendor\_id())){

return new Page<CommExchange>(0);

}

exchange.setVendor\_id(Integer.parseInt(logUser.getVendor\_id()));

return exchangeService.getPageList(BeanUtils.convert2Map(exchange), start, limit);

}

/\*\*

\* 公益礼品领取

\* @param id

\* @return

\*/

@RequestMapping("/takeGift")

@ResponseBody

public ResponseData takeGift(Integer id){

CommExchange exchange = exchangeService.findEntity(id);

if(exchange.getState() == 0){

exchangeService.updateEntity(exchange);

}

return ResponseData.SUCCESS\_NO\_DATA;

}

}

package com.yawa.busi.controller;

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

import org.springframework.stereotype.Controller;

import org.springframework.util.StringUtils;

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

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

import com.yawa.core.model.User;

import com.yawa.core.service.UserService;

import com.yawa.http.model.CommExchange;

import com.yawa.http.service.ExchangeService;

import com.yawa.util.BeanUtils;

import com.yawa.util.context.SecurityContextUtil;

import com.yawa.util.model.Page;

import com.yawa.util.model.ResponseData;

@Controller

@RequestMapping("/exchange")

public class ExchangeController {

@Autowired

private ExchangeService exchangeService;

@Autowired

private UserService userService;

/\*\*

\* 分页查询列表

\* @param start

\* @param limit

\* @param exchange

\* @return

\*/

@RequestMapping("/getPageList")

@ResponseBody

public Page<CommExchange> getPageList(int start,int limit,CommExchange exchange){

Integer user\_id = SecurityContextUtil.getCurrentUser().getId();

User logUser = this.userService.findEntity(user\_id);

if(StringUtils.isEmpty(logUser.getVendor\_id())){

return new Page<CommExchange>(0);

}

exchange.setVendor\_id(Integer.parseInt(logUser.getVendor\_id()));

return exchangeService.getPageList(BeanUtils.convert2Map(exchange), start, limit);

}

/\*\*

\* 公益礼品领取

\* @param id

\* @return

\*/

@RequestMapping("/takeGift")

@ResponseBody

public ResponseData takeGift(Integer id){

CommExchange exchange = exchangeService.findEntity(id);

if(exchange.getState() == 0){

exchangeService.updateEntity(exchange);

}

return ResponseData.SUCCESS\_NO\_DATA;

}

}

package com.yawa.busi.controller;

import java.io.IOException;

import java.util.HashMap;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.apache.commons.lang.StringUtils;

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.ResponseBody;

import com.yawa.busi.model.IncApply;

import com.yawa.busi.service.IncApplyService;

import com.yawa.busi.util.ConstantItems;

import com.yawa.util.attach.Asset;

import com.yawa.util.attach.AssetHelper;

import com.yawa.util.context.SecurityContextUtil;

import com.yawa.util.file.FileUpload;

import com.yawa.util.model.Page;

import com.yawa.util.model.ResponseData;

@Controller

@RequestMapping("/apply")

public class IncApplyController {

@Autowired

private IncApplyService incApplyService;

@Autowired

private AssetHelper assetHelper;

@RequestMapping(value = "/getIncApplyPage")

@ResponseBody

public Page<IncApply> getIncApplyPage(int start,int limit,String state,String keyWords){

Map<String,Object> params = new HashMap<String, Object>();

params.put("vendor\_id", Integer.parseInt(SecurityContextUtil.getLoginVendorId()));

params.put("state", state);

params.put("keyWords",keyWords);

return incApplyService.getPageList(params, start, limit);

}

@RequestMapping(value = "/updateApplyState")

@ResponseBody

public ResponseData updateApplyState(String operType,String ids){

if(StringUtils.isNotEmpty(operType) && StringUtils.isNotEmpty(ids)){

incApplyService.updateApplyState(operType,ids);

}

return ResponseData.SUCCESS\_NO\_DATA;

}

@RequestMapping(value = "/loadApplyById")

@ResponseBody

public IncApply loadApplyById(Integer id){

IncApply incApply = new IncApply();

if(id != null){

incApply = incApplyService.findEntity(id);

Asset asset = assetHelper.getAssetByViewable(ConstantItems.CONST\_MAP.get("VIEABLE\_IMG\_APPLY"), id);

incApply.setAsset(asset);

}

return incApply;

}

@RequestMapping(value = "/updateApply")

public void updateApply(HttpServletRequest request,HttpServletResponse response,IncApply incApply){

try {

new FileUpload<IncApply>().uploadFile(incApply,request);

} catch (Exception e) {}

incApply.setVendor\_id(Integer.parseInt(SecurityContextUtil.getLoginVendorId()));

if(incApply.getId() == null){

incApplyService.insertEntity(incApply);

}else{

incApplyService.updateEntity(incApply);

}

Asset asset = incApply.getAsset();

asset.setViewable\_id(incApply.getId());

asset.setViewable\_type(ConstantItems.CONST\_MAP.get("VIEABLE\_IMG\_APPLY"));

asset.setMedium\_size(incApply.getSpec().trim());

assetHelper.init();

assetHelper.saveAsset(asset);

response.setContentType("text/html;charset=UTF-8");

response.setHeader("Pragma", "No-cache");

response.setHeader("Cache-Control", "no-cache");

response.setDateHeader("Expires", 0);

try {

response.getWriter().write("{success: true}");

} catch (IOException e) {

e.printStackTrace();

}

}

}

package com.yawa.busi.controller;

import java.io.IOException;

import java.util.HashMap;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.apache.commons.lang.StringUtils;

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.ResponseBody;

import com.yawa.busi.model.IncNews;

import com.yawa.busi.service.IncNewsService;

import com.yawa.busi.util.ConstantItems;

import com.yawa.util.attach.Asset;

import com.yawa.util.attach.AssetHelper;

import com.yawa.util.context.SecurityContextUtil;

import com.yawa.util.file.FileUpload;

import com.yawa.util.model.Page;

import com.yawa.util.model.ResponseData;

@Controller

@RequestMapping("/governmentManager")

public class IncNewsController {

@Autowired

private IncNewsService incNewsService;

@Autowired

private AssetHelper assetHelper;

/\*\*

\* 政府办事分页列表

\* @param start

\* @param limit

\* @param source

\* @return

\*/

@RequestMapping("/queryList")

@ResponseBody

public Page<IncNews> getList(int start,int limit,String source){

Map<String,Object> params = new HashMap<String, Object>();

params.put("source",source);

return incNewsService.getPageList(params, start, limit);

}

/\*\*

\* 根据ID获取消息内容

\* @param id

\* @return

\*/

@RequestMapping(value = "/loadApplyById")

@ResponseBody

public IncNews loadApplyById(Integer id) {

return incNewsService.findEntity(id);

}

//即时通的修改状态

@RequestMapping(value = "/updateApplyState")

@ResponseBody

public ResponseData updateApplyState(String operType, String ids) {

if (StringUtils.isNotEmpty(operType) && StringUtils.isNotEmpty(ids)) {

incNewsService.updateApplyState(operType, ids);

}

return ResponseData.SUCCESS\_NO\_DATA;

}

//即时通的添加及修改

@RequestMapping(value = "/updateApply")

public void updateApply(HttpServletRequest request,

HttpServletResponse response, IncNews incNews) {

try {

new FileUpload<IncNews>().uploadFile(incNews, request);

} catch (Exception e) {}

incNews.setVendor\_id(Integer.parseInt(SecurityContextUtil.getLoginVendorId()));

incNews.setUser\_id(SecurityContextUtil.getLoginUserId() + "");

incNews.setUser\_name(SecurityContextUtil.getCurrentUser().getLogin\_name());

incNews.setSource("3");

incNews.setChannel("2");

incNews.setType\_id("3");

incNews.setReceive\_type("2");

incNews.setStatus("0");

if (incNews.getId()==0) {

incNewsService.insertEntity(incNews);

} else {

incNewsService.updateApply(incNews);

}

Asset asset = incNews.getAttach();

asset.setViewable\_id(incNews.getId());

asset.setViewable\_type(ConstantItems.CONST\_MAP

.get("VIEABLE\_IMG\_NEWS"));

assetHelper.init();

assetHelper.saveAsset(asset);

response.setContentType("text/html;charset=UTF-8");

response.setHeader("Pragma", "No-cache");

response.setHeader("Cache-Control", "no-cache");

response.setDateHeader("Expires", 0);

try {

response.getWriter().write("{success: true}");

} catch (IOException e) {

e.printStackTrace();

}

}

}

package com.yawa.busi.controller;

import java.text.SimpleDateFormat;

import java.util.Calendar;

import java.util.HashMap;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

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.ResponseBody;

import com.yawa.busi.model.Order;

import com.yawa.busi.model.OrderStatis;

import com.yawa.busi.service.OrderStatisService;

import com.yawa.core.model.User;

import com.yawa.util.model.Page;

/\*\*

\* 订单

\* @author am

\* @since 2014-09-16

\* @desc

\*/

@Controller

@RequestMapping("/orderStatis")

public class OrderStatisController {

@Autowired

private OrderStatisService orderStatisService;

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd");

@RequestMapping("/queryList")

@ResponseBody

public Page<OrderStatis> getList(HttpServletRequest request,int start,int limit,String starttime,String endtime,int cummunityid){

User user = (User) request.getSession().getAttribute("currentUser");

Map<String,Object> params = new HashMap<String, Object>();

params.put("starttime", starttime);

params.put("endtime", endtime);

params.put("community\_id", cummunityid);

params.put("vendor\_id",Integer.parseInt(user.getVendor\_id()));

return orderStatisService.getPageList(params, start, limit);

}

@RequestMapping("/getordercm")

@ResponseBody

public Map<String,Object> getordercountandmoney(HttpServletRequest request){

User user = (User) request.getSession().getAttribute("currentUser");

Map<String,Object> params = new HashMap<String, Object>();

params.put("vendor\_id",Integer.parseInt(user.getVendor\_id()));

return orderStatisService.getordercountandmoney(params);

}

//柱状图----周

@RequestMapping("/getorderweek")

@ResponseBody

public Page<Order> getorderweekmoney(HttpServletRequest request,String type){

User user = (User) request.getSession().getAttribute("currentUser");

Map<String,Object> params = new HashMap<String, Object>();

params.put("vendor\_id",Integer.parseInt(user.getVendor\_id()));

Calendar c = Calendar.getInstance();

// 默认时，每周第一天为星期日，需要更改一下

c = Calendar.getInstance();

c.setFirstDayOfWeek(Calendar.MONDAY);

c.set(Calendar.DAY\_OF\_WEEK, Calendar.MONDAY);

String first = format.format(c.getTime());

c.set(Calendar.DAY\_OF\_WEEK, Calendar.SUNDAY);

String last = format.format(c.getTime());

params.put("first", first);

params.put("last", last);

return orderStatisService.getorderweekmoney(params,0,6);

}

//环形图

@RequestMapping("/getorderweekBycateg")

@ResponseBody

public Map<String,Object> getordercountandmoneyBycateg(HttpServletRequest request,Integer categoryid,String type){

User user = (User) request.getSession().getAttribute("currentUser");

Map<String,Object> params = new HashMap<String, Object>();

params.put("vendor\_id",Integer.parseInt(user.getVendor\_id()));

params.put("category\_id",categoryid);

if(type.equals("week")){

// 默认时，每周第一天为星期日，需要更改一下

Calendar c = Calendar.getInstance();

c = Calendar.getInstance();

c.setFirstDayOfWeek(Calendar.MONDAY);

c.set(Calendar.DAY\_OF\_WEEK, Calendar.MONDAY);

String first = format.format(c.getTime());

c.set(Calendar.DAY\_OF\_WEEK, Calendar.SUNDAY);

String last = format.format(c.getTime());

params.put("first", first);

params.put("last", last);

}else if(type.equals("month")){

Calendar c = Calendar.getInstance();

c.add(Calendar.MONTH, 0);

c.set(Calendar.DAY\_OF\_MONTH,1);//设置为1号,当前日期既为本月第一天

String first = format.format(c.getTime());

//获取当前月最后一天

Calendar ca = Calendar.getInstance();

ca.set(Calendar.DAY\_OF\_MONTH, ca.getActualMaximum(Calendar.DAY\_OF\_MONTH));

String last = format.format(ca.getTime());

params.put("first", first);

params.put("last", last);

}

return orderStatisService.getordercountandmoneyBycateg(params);

}

//获得所有的分类业务

@RequestMapping("/getallcateg")

@ResponseBody

public Page<Order> getallcateg(HttpServletRequest request,String type){

User user = (User) request.getSession().getAttribute("currentUser");

Map<String,Object> params = new HashMap<String, Object>();

params.put("vendor\_id",Integer.parseInt(user.getVendor\_id()));

if(type.equals("week")){

Calendar c = Calendar.getInstance();

// 默认时，每周第一天为星期日，需要更改一下

c = Calendar.getInstance();

c.setFirstDayOfWeek(Calendar.MONDAY);

c.set(Calendar.DAY\_OF\_WEEK, Calendar.MONDAY);

String first = format.format(c.getTime());

c.set(Calendar.DAY\_OF\_WEEK, Calendar.SUNDAY);

String last = format.format(c.getTime());

params.put("first", first);

params.put("last", last);

}else if(type.equals("month")){

Calendar c = Calendar.getInstance();

c.add(Calendar.MONTH, 0);

c.set(Calendar.DAY\_OF\_MONTH,1);//设置为1号,当前日期既为本月第一天

String first = format.format(c.getTime());

//获取当前月最后一天

Calendar ca = Calendar.getInstance();

ca.set(Calendar.DAY\_OF\_MONTH, ca.getActualMaximum(Calendar.DAY\_OF\_MONTH));

String last = format.format(ca.getTime());

params.put("first", first);

params.put("last", last);

}

return orderStatisService.getallcateg(params,0,10);

}

//得到最大的业务分类

@RequestMapping("/getallcategmax")

@ResponseBody

public Map<String,Object> getallcategmax(HttpServletRequest request,String type){

User user = (User) request.getSession().getAttribute("currentUser");

Map<String,Object> params = new HashMap<String, Object>();

params.put("vendor\_id",Integer.parseInt(user.getVendor\_id()));

if(type.equals("week")){

// 默认时，每周第一天为星期日，需要更改一下

Calendar c = Calendar.getInstance();

c = Calendar.getInstance();

c.setFirstDayOfWeek(Calendar.MONDAY);

c.set(Calendar.DAY\_OF\_WEEK, Calendar.MONDAY);

String first = format.format(c.getTime());

c.set(Calendar.DAY\_OF\_WEEK, Calendar.SUNDAY);

String last = format.format(c.getTime());

params.put("first", first);

params.put("last", last);

}else if(type.equals("month")){

Calendar c = Calendar.getInstance();

c.add(Calendar.MONTH, 0);

c.set(Calendar.DAY\_OF\_MONTH,1);//设置为1号,当前日期既为本月第一天

String first = format.format(c.getTime());

//获取当前月最后一天

Calendar ca = Calendar.getInstance();

ca.set(Calendar.DAY\_OF\_MONTH, ca.getActualMaximum(Calendar.DAY\_OF\_MONTH));

String last = format.format(ca.getTime());

params.put("first", first);

params.put("last", last);

}

return orderStatisService.getallcategmax(params);

}

//得到最大的业务分类

@RequestMapping("/getallcategmin")

@ResponseBody

public Map<String,Object> getallcategmin(HttpServletRequest request,String type){

User user = (User) request.getSession().getAttribute("currentUser");

Map<String,Object> params = new HashMap<String, Object>();

params.put("vendor\_id",Integer.parseInt(user.getVendor\_id()));

if(type.equals("week")){

Calendar c = Calendar.getInstance();

// 默认时，每周第一天为星期日，需要更改一下

c = Calendar.getInstance();

c.setFirstDayOfWeek(Calendar.MONDAY);

c.set(Calendar.DAY\_OF\_WEEK, Calendar.MONDAY);

String first = format.format(c.getTime());

c.set(Calendar.DAY\_OF\_WEEK, Calendar.SUNDAY);

String last = format.format(c.getTime());

params.put("first", first);

params.put("last", last);

}else if(type.equals("month")){

Calendar c = Calendar.getInstance();

c.add(Calendar.MONTH, 0);

c.set(Calendar.DAY\_OF\_MONTH,1);//设置为1号,当前日期既为本月第一天

String first = format.format(c.getTime());

//获取当前月最后一天

Calendar ca = Calendar.getInstance();

ca.set(Calendar.DAY\_OF\_MONTH, ca.getActualMaximum(Calendar.DAY\_OF\_MONTH));

String last = format.format(ca.getTime());

params.put("first", first);

params.put("last", last);

}

return orderStatisService.getallcategmin(params);

}

//柱状图---月

@RequestMapping("/getordemonth")

@ResponseBody

public Page<Order> getordemonth(HttpServletRequest request,String type){

User user = (User) request.getSession().getAttribute("currentUser");

Map<String,Object> params = new HashMap<String, Object>();

Calendar caw = Calendar.getInstance();

params.put("vendor\_id",Integer.parseInt(user.getVendor\_id()));

caw.add(Calendar.MONTH, 0);

caw.set(Calendar.DAY\_OF\_MONTH,1);//设置为1号,当前日期既为本月第一天

String first = format.format(caw.getTime());

//获取当前月最后一天

Calendar ca = Calendar.getInstance();

ca.set(Calendar.DAY\_OF\_MONTH, ca.getActualMaximum(Calendar.DAY\_OF\_MONTH));

String last = format.format(ca.getTime());

params.put("first", first);

params.put("last", last);

return orderStatisService.getordemonth(params,0,31);

}

}

package com.yawa.busi.controller;

import java.io.File;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

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

import org.springframework.stereotype.Controller;

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

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

import com.yawa.busi.model.AuditingLog;

import com.yawa.busi.model.CommunityVO;

import com.yawa.busi.model.ProductsVO;

import com.yawa.busi.service.AuditingLogService;

import com.yawa.busi.service.ProjectServiceService;

import com.yawa.busi.service.ServiceDistributionService;

import com.yawa.busi.util.ConstantItems;

import com.yawa.core.model.User;

import com.yawa.core.service.UserService;

import com.yawa.util.BeanUtils;

import com.yawa.util.attach.Asset;

import com.yawa.util.attach.AssetHelper;

import com.yawa.util.context.SecurityContextUtil;

import com.yawa.util.file.FileUpload;

import com.yawa.util.model.Page;

import com.yawa.util.model.ResponseData;

@Controller

@RequestMapping("/projectService")

public class ProjectServiceController {

@Autowired

private ProjectServiceService projectServiceService;

@Autowired

private ServiceDistributionService serviceDistributionService;

@Autowired

private AssetHelper assetHelper;

@Autowired

private UserService userService;

@Autowired

private AuditingLogService logService;

@RequestMapping("/queryList")

@ResponseBody

public Page<ProductsVO> getList(int start,int limit,int vendor\_id,String name,String searchCategories){

List<CommunityVO> communityList = this.serviceDistributionService.getCommunityList(vendor\_id);

String community\_name = "";

for(int i=0;i<communityList.size();i++){

community\_name += ","+communityList.get(i).getCommunity\_name();

}

Map<String,Object> params = new HashMap<String, Object>();

if(!searchCategories.equals("all")&&!searchCategories.equals("")){

params.put("categories", Integer.parseInt(searchCategories));

}

params.put("product\_name", name);

params.put("vendor\_id",vendor\_id);

if(community\_name.equals("")){

params.put("community\_name",community\_name);

}else{

params.put("community\_name",community\_name.substring(1, community\_name.length()));

}

Page<ProductsVO> page = projectServiceService.getPageList(params, start, limit);

return page;

}

/\*\*

\* 公益礼品分页查询

\* @param start

\* @param limit

\* @param product

\* @return

\*/

@RequestMapping("/getPageList")

@ResponseBody

public Page<ProductsVO> getPageList(HttpServletRequest request,int start,int limit,ProductsVO product){

Integer user\_id = SecurityContextUtil.getCurrentUser().getId();

User logUser = this.userService.findEntity(user\_id);

product.setVendor\_id(logUser.getVendor\_id());

return projectServiceService.getBenefitPageList(BeanUtils.convert2Map(product), start, limit);

}

@RequestMapping("/insert")

public String insert(HttpServletRequest request,HttpServletResponse response){

ProductsVO product = new ProductsVO();

try {

new FileUpload<ProductsVO>().uploadFile(product,request);

} catch (Exception e) {

e.printStackTrace();

}

User logUser = this.userService.findEntity(SecurityContextUtil.getCurrentUser().getId());

AuditingLog log = new AuditingLog();

log.setAuditor(logUser.getId().toString());

log.setAuditor\_name(logUser.getLogin\_name());

log.setCategory("gysh");

if(product.getId() == null){

product.setExchange("0");

product.setVendor\_id(logUser.getVendor\_id());

log.setBusiid(product.getId());

this.projectServiceService.insertProduct(product);

if(product.getState().equals("D")){

log.setRemarks("新建公益礼品，未上架");

logService.insertEntity(log);

}else if(product.getState().equals("A")){

log.setRemarks("新建公益礼品，上架审核");

logService.insertEntity(log);

}

}else{

this.projectServiceService.updateEntity(product);

log.setBusiid(product.getId());

if(product.getState().equals("D")){

log.setRemarks("修改公益礼品，未上架");

logService.insertEntity(log);

}else if(product.getState().equals("A")){

log.setRemarks("修改公益礼品，上架审核");

logService.insertEntity(log);

}

}

//上传附件

Asset asset = product.getAttach();

if(asset != null && asset.getUploadFile() != null){

asset.setViewable\_id(product.getId());

asset.setViewable\_type(ConstantItems.CONST\_MAP.get("PRODUCT\_IMG\_REG"));

asset.setImageCutPos(product.getImageX() + ":" + product.getImageY() + ":" + product.getImageW() + ":" + product.getImageH());

assetHelper.init();

assetHelper.saveAsset(asset);

}

return "redirect:/loginManager/toDistrMainPage";

}

@RequestMapping("/getById")

@ResponseBody

public ProductsVO getById(int id){

ProductsVO p = new ProductsVO();

p = this.projectServiceService.getById(id);

Asset asset = this.assetHelper.getAssetByViewable(ConstantItems.CONST\_MAP.get("PRODUCT\_IMG\_REG"), id);

if(asset!=null){

File file = this.assetHelper.getStoreFileDir(asset,ConstantItems.CONST\_MAP.get("ASSET\_MEDIUM"));

if(file.exists()){

asset.setUploadFile(file);

}

}

p.setAttach(asset);

return p;

}

@RequestMapping("/deleteById")

@ResponseBody

public ProductsVO deleteById(int id){

ProductsVO p = new ProductsVO();

p.setState("003");//删除状态

p.setId(id);

updateState(p);

return p;

}

/\*\*

\* 公益礼品操作

\* @param id

\* @param state

\* @return

\*/

@RequestMapping("/productAction")

@ResponseBody

public ResponseData productAction(int id,String state){

ProductsVO product = this.projectServiceService.getById(id);

if(product != null){

if(product.getState().equals("D") && state.equals("U")){

return new ResponseData(false,"此产品尚未上架！");

}else if(product.getState().equals("A") && state.equals("U")){

return new ResponseData(false,"此产品尚未上架！");

}else if(product.getState().equals("A") && state.equals("003")){

return new ResponseData(false,"此产品正在上架中禁止删除！");

}else if(product.getState().equals("F") && !state.equals("U")){

return new ResponseData(false,"此产品已经上架！");

}

User logUser = this.userService.findEntity(SecurityContextUtil.getCurrentUser().getId());

AuditingLog log = new AuditingLog();

log.setAuditor(logUser.getId().toString());

log.setAuditor\_name(logUser.getLogin\_name());

log.setCategory("gysh");

log.setBusiid(product.getId());

product.setState(state);

projectServiceService.updateState(product);

if(state.equals("A")){

log.setRemarks("礼品上架审核");

}else if(state.equals("D")){

log.setRemarks("礼品上架撤回");

}else if(state.equals("U")){

log.setRemarks("礼品下架");

}else if(state.equals("003")){

log.setRemarks("礼品删除");

}

logService.insertEntity(log);

}

return ResponseData.SUCCESS\_NO\_DATA;

}

public void updateState(ProductsVO p){

this.projectServiceService.updateState(p);

}

}

package com.yawa.busi.controller;

import java.io.File;

import java.io.IOException;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

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

import org.springframework.stereotype.Controller;

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

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

import com.yawa.app.service.AppMessageService;

import com.yawa.busi.model.CategoriesVO;

import com.yawa.busi.model.ClassificationsVO;

import com.yawa.busi.model.CommunityVO;

import com.yawa.busi.model.IncNews;

import com.yawa.busi.model.VendorVO;

import com.yawa.busi.service.CommunityService;

import com.yawa.busi.service.IncNewsService;

import com.yawa.busi.service.ServiceDistributionService;

import com.yawa.busi.util.ConstantItems;

import com.yawa.core.model.User;

import com.yawa.core.service.UserService;

import com.yawa.util.attach.Asset;

import com.yawa.util.attach.AssetHelper;

import com.yawa.util.context.SecurityContextUtil;

import com.yawa.util.file.FileUpload;

import com.yawa.util.model.Page;

@Controller

@RequestMapping("/serviceDistribution")

public class ServiceDistributionController {

@Autowired

private ServiceDistributionService serviceDistributionService;

@Autowired

private CommunityService communityService;

// 向四川APP写入商超数据

@Autowired

private AppMessageService appMessageService;

@Autowired

private AssetHelper assetHelper;

@Autowired

private UserService userService;

@Autowired

private IncNewsService incNewsService;

@RequestMapping("/insert")

public void insert(HttpServletRequest request,

HttpServletResponse response) {

response.setContentType("text/html;charset=UTF-8");

response.setHeader("Pragma", "No-cache");

response.setHeader("Cache-Control", "no-cache");

response.setDateHeader("Expires", 0);

VendorVO vendorVO = new VendorVO();

try {

new FileUpload<VendorVO>().uploadFile(vendorVO, request);

} catch (Exception e) {

try {

response.getWriter().write("{success: false}");

} catch (IOException e1) {

e1.printStackTrace();

}

}

try{

User user = (User) request.getSession().getAttribute(

SecurityContextUtil.SPRING\_SESSION\_LOGIN\_USER\_KEY);

// 新增商超主表信息

this.serviceDistributionService.insertVendor(vendorVO);

// 新增商超服务范围信息

this.serviceDistributionService.insertCommunitiesVendors(vendorVO);

if (vendorVO.getGroup\_id() == 2) {

// 新增商超配送费用信息

this.serviceDistributionService.insertFee(vendorVO);

}

// 新增商超经营范围信息

this.serviceDistributionService.insertCategories(vendorVO);

// 上传附件

Asset asset = vendorVO.getAttach();

asset.setViewable\_id(vendorVO.getId());

asset.setViewable\_type(ConstantItems.CONST\_MAP.get("VIEWABLE\_TYPE\_REG"));

assetHelper.init();

assetHelper.saveAsset(asset);

// 修改登录人的所属商超

this.serviceDistributionService.updateUser(user.getId(),vendorVO.getId());

user = this.userService.findUser(user);

request.getSession().removeAttribute(

SecurityContextUtil.SPRING\_SESSION\_LOGIN\_USER\_KEY);// 移除用户，但session不变，下次登陆的时候看到的sessionID还是一样的

request.getSession().setAttribute(

SecurityContextUtil.SPRING\_SESSION\_LOGIN\_USER\_KEY, user);

// 上门服务 APP新增

if (vendorVO.getGroup\_id() == Integer.parseInt(ConstantItems.CONST\_MAP.get("GROUP\_DOOR"))) {

appMessageService.updateVendor(vendorVO);

}

}catch(Exception e){

try {

response.getWriter().write("{success: false}");

} catch (IOException e1) {

e1.printStackTrace();

}

}

try {

response.getWriter().write("{success: true}");

} catch (IOException e) {

e.printStackTrace();

}

}

@RequestMapping("/update")

public void update(HttpServletRequest request,

HttpServletResponse response) {

response.setContentType("text/html;charset=UTF-8");

response.setHeader("Pragma", "No-cache");

response.setHeader("Cache-Control", "no-cache");

response.setDateHeader("Expires", 0);

VendorVO vendorVO = new VendorVO();

try {

new FileUpload<VendorVO>().uploadFile(vendorVO, request);

} catch (Exception e) {

try {

response.getWriter().write("{success: false}");

} catch (IOException e1) {

e1.printStackTrace();

}

}

try{

// 修改商超主表信息

this.serviceDistributionService.updateEntity(vendorVO);

// 删除商超服务范围信息

this.serviceDistributionService.deleteCommunitiesVendors(vendorVO);

// 新增商超服务范围信息

this.serviceDistributionService.insertCommunitiesVendors(vendorVO);

// 删除商超经营范围删除的信息

this.serviceDistributionService.deleteCategories(vendorVO);

// 新增商超经营范围新增的信息

this.serviceDistributionService.insertElseCategories(vendorVO);

// 上传修改头像

Asset asset = vendorVO.getPic();

asset.setViewable\_id(vendorVO.getId());

asset.setViewable\_type(ConstantItems.CONST\_MAP.get("VIEWABLE\_HEAD\_REG"));

assetHelper.init();

assetHelper.saveAsset(asset);

// 修改营业执照

Asset asset1 = vendorVO.getAttach();

asset1.setViewable\_id(vendorVO.getId());

asset1.setViewable\_type(ConstantItems.CONST\_MAP

.get("VIEWABLE\_TYPE\_REG"));

assetHelper.saveAsset(asset1);

// 上门服务 APP新增

if (SecurityContextUtil.getLoginGroup().equals(ConstantItems.CONST\_MAP.get("GROUP\_DOOR"))){

appMessageService.updateVendor(vendorVO);

}

}catch(Exception e){

try {

response.getWriter().write("{success: false}");

} catch (IOException e1) {

e1.printStackTrace();

}

}

try {

User user = (User) request.getSession().getAttribute(

SecurityContextUtil.SPRING\_SESSION\_LOGIN\_USER\_KEY);

user = this.userService.findUser(user);

request.getSession().removeAttribute(

SecurityContextUtil.SPRING\_SESSION\_LOGIN\_USER\_KEY);// 移除用户，但session不变，下次登陆的时候看到的sessionID还是一样的

request.getSession().setAttribute(

SecurityContextUtil.SPRING\_SESSION\_LOGIN\_USER\_KEY, user);

response.getWriter().write("{success: true}");

} catch (IOException e) {

e.printStackTrace();

}

}

/\*\*

\* 获取社区列表

\*

\* @return

\*/

@RequestMapping(value = "/getCommunityList")

@ResponseBody

public List<CommunityVO> getCommunityList() {

CommunityVO c = new CommunityVO();

return this.communityService.getList(c);

}

/\*\*

\* 根据分组id获取所有经营分类（从前台获取分组id）

\*

\* @param groupId

\* @return

\*/

@RequestMapping(value = "/getAllCategoriesListByGroupId")

@ResponseBody

public List<ClassificationsVO> getAllCategoriesListByGroupId(int groupId) {

List<ClassificationsVO> list = this.serviceDistributionService

.getAllCategoriesList(groupId);

return list;

}

/\*\*

\* 根据分组id获取所有经营分类（从当前登录人获取分组id）

\*

\* @param request

\* @return

\*/

@RequestMapping(value = "/getAllCategoriesList")

@ResponseBody

public List<ClassificationsVO> getAllCategoriesList(

HttpServletRequest request) {

User user = (User) request.getSession().getAttribute("currentUser");

int groupId = Integer.parseInt(user.getGroup\_id());

List<ClassificationsVO> list = this.serviceDistributionService

.getAllCategoriesList(groupId);

return list;

}

/\*\*

\* 根据商超id获取商超信息

\*

\* @param id

\* @return

\*/

@RequestMapping(value = "/getVendorsById")

@ResponseBody

public Map<String,Object> getVendorsById(HttpServletRequest request) {

User user = SecurityContextUtil.getCurrentUser();

int id = Integer.parseInt(user.getVendor\_id());

VendorVO v = new VendorVO();

v.setId(id);

v = this.serviceDistributionService.getList(v).get(0);

Asset pic = this.assetHelper.getAssetByViewable(

ConstantItems.CONST\_MAP.get("VIEWABLE\_HEAD\_REG"), id);

if(pic!=null){

File file = this.assetHelper.getStoreFileDir(pic,ConstantItems.CONST\_MAP.get("ASSET\_ORIGIN"));

if(file.exists()){

pic.setUploadFile(file);

}

}

v.setPic(pic);

Map<String,Object> params = new HashMap<String, Object>();

params.put("source","1");

Page<IncNews> page = incNewsService.getPageList(params, 0, 1);

Map<String,Object> map = new HashMap<String,Object>();

map.put("vendorVO", v);

map.put("incNews", page);

return map;

}

/\*\*

\* 根据商超id获取商超信息

\*

\* @param id

\* @return

\*/

@RequestMapping(value = "/getPersonnalList")

@ResponseBody

public Map<String,Object> getPersonnalList(HttpServletRequest request,int id) {

VendorVO v = new VendorVO();

v.setId(id);

v = this.serviceDistributionService.getList(v).get(0);

Asset attach = this.assetHelper.getAssetByViewable(

ConstantItems.CONST\_MAP.get("VIEWABLE\_TYPE\_REG"), id);

if(attach!=null){

File file = this.assetHelper.getStoreFileDir(attach,ConstantItems.CONST\_MAP.get("ASSET\_MEDIUM"));

if(file.exists()){

attach.setUploadFile(file);

}

}

v.setAttach(attach);

String community\_name = "";

String community\_id = "";

List<CommunityVO> list = this.serviceDistributionService

.getCommunityList(id);

for (int i = 0; i < list.size(); i++) {

community\_name += "," + list.get(i).getCommunity\_name();

community\_id += "," + list.get(i).getId();

}

v.setCommunity(community\_id + ",");

v.setCommunity\_name(community\_name + ",");

List<CategoriesVO> list1 = this.serviceDistributionService

.getCategoriesList(id);

String categories = "";

String categories\_name = "";

for (int i = 0; i < list1.size(); i++) {

categories += "," + list1.get(i).getClassification\_id();

categories\_name += "," + list1.get(i).getCategory\_name();

}

v.setCategories(categories + ",");

v.setCategories\_name(categories\_name + ",");

Asset attach1 = this.assetHelper.getAssetByViewable(

ConstantItems.CONST\_MAP.get("VIEWABLE\_HEAD\_REG"), id);

if(attach1!=null){

File file1= this.assetHelper.getStoreFileDir(attach1,ConstantItems.CONST\_MAP.get("ASSET\_MEDIUM"));

if(file1.exists()){

attach1.setUploadFile(file1);

}

}

v.setPic(attach1);

Map<String,Object> map = new HashMap<String,Object>();

map.put("vendorVO", v);

//获取该商超商品所覆盖的商品类型

List<CategoriesVO> projectCategoriesList = getProductsCategoryList(id);

map.put("projectCategoriesList", projectCategoriesList);

//所有社区列表

List<CommunityVO> communityList = getCommunityList();

map.put("communityList", communityList);

//根据入住分类获取所有分类列表

User user = (User) request.getSession().getAttribute("currentUser");

int groupId = Integer.parseInt(user.getGroup\_id());

List<ClassificationsVO> classificationsList = getAllCategoriesListByGroupId(groupId);

map.put("classificationsVOList", classificationsList);

return map;

}

/\*\*

\* 获取商超的经营范围

\*

\* @param id

\* @return

\*/

@RequestMapping(value = "/getCategoriesList")

@ResponseBody

public List<CategoriesVO> getCategoriesList(HttpServletRequest request) {

User user = SecurityContextUtil.getCurrentUser();

int id = Integer.parseInt(user.getVendor\_id());

List<CategoriesVO> list = this.serviceDistributionService

.getCategoriesList(id);

return list;

}

/\*\*

\* 获取商超的商品所属类型列表

\*

\* @param vendor\_id

\* @return

\*/

@RequestMapping(value = "/getProductsCategoryList")

@ResponseBody

public List<CategoriesVO> getProductsCategoryList(int vendor\_id) {

List<CategoriesVO> list = this.serviceDistributionService

.getProductsCategoryList(vendor\_id);

return list;

}

}

package com.yawa.busi.controller;

import java.util.HashMap;

import java.util.Map;

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.ResponseBody;

import com.yawa.busi.model.TellApply;

import com.yawa.busi.service.TellApplyService;

import com.yawa.util.context.SecurityContextUtil;

import com.yawa.util.model.Page;

@Controller

@RequestMapping("/tellapply")

public class TellApplyController {

@Autowired

private TellApplyService tellApplyService;

@RequestMapping(value = "/getIncApplyPage")

@ResponseBody

public Page<TellApply> getIncApplyPage(int start, int limit, String state,

String keyWords) {

Map<String, Object> params = new HashMap<String, Object>();

params.put("vendor\_id",

Integer.parseInt(SecurityContextUtil.getLoginVendorId()));

params.put("status", state);

params.put("keyWords", keyWords);

return tellApplyService.getPageList(params, start, limit);

}

}

package com.yawa.busi.service;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

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

import org.springframework.stereotype.Service;

import com.yawa.base.dao.GenericDAO;

import com.yawa.base.service.BaseService;

import com.yawa.busi.dao.ServiceDistributionDao;

import com.yawa.busi.model.CategoriesVO;

import com.yawa.busi.model.ClassificationsVO;

import com.yawa.busi.model.CommunityVO;

import com.yawa.busi.model.VendorVO;

@Service

public class ServiceDistributionService extends BaseService<VendorVO, Integer> {

@Autowired

private ServiceDistributionDao serviceDistributionDao;

@Override

public GenericDAO<VendorVO, Integer> getGenericDao() {

return this.serviceDistributionDao;

}

public void insertVendor(VendorVO vendorVO) {

this.serviceDistributionDao.insertVendor(vendorVO);

}

public void insertCommunitiesVendors(VendorVO vendorVO) {

Map<String, Object> paramMap = new HashMap<String, Object>();

int vendor\_id = vendorVO.getId();

paramMap.put("vendor\_id", vendor\_id);

String[] community = vendorVO.getCommunity().split(",");

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

String community\_id = community[i];

if (!community\_id.equals("")) {

paramMap.put("community\_id", community\_id);

this.serviceDistributionDao.insertCommunitiesVendors(paramMap);

}

}

}

public void insertCategories(VendorVO vendorVO) {

int vendor\_id = vendorVO.getId();

String[] categories = vendorVO.getCategories().split(",");

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

Map<String, Object> paramMap = new HashMap<String, Object>();

paramMap.put("vendor\_id", vendor\_id);

String category\_id = categories[i];

if (!category\_id.equals("")) {

paramMap.put("category\_id", Integer.parseInt(category\_id));

this.serviceDistributionDao.insertCategories(paramMap);

}

}

}

public void insertElseCategories(VendorVO vendorVO) {

int vendor\_id = vendorVO.getId();

String[] categories = vendorVO.getCategories()

.substring(1, vendorVO.getCategories().length()).split(",");

Map<String, Object> paramMap = new HashMap<String, Object>();

paramMap.put("vendor\_id", vendor\_id);

Integer categories\_id[] = new Integer[categories.length];

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

if (!categories[i].equals("")) {

categories\_id[i] = Integer.parseInt(categories[i]);

}

}

paramMap.put("categories\_id", categories\_id);

this.serviceDistributionDao.insertElseCategories(paramMap);

}

public void deleteCommunitiesVendors(VendorVO vendorVO) {

Map<String, Object> paramMap = new HashMap<String, Object>();

int vendor\_id = vendorVO.getId();

paramMap.put("vendor\_id", vendor\_id);

this.serviceDistributionDao.deleteCommunitiesVendors(paramMap);

}

public void deleteCategories(VendorVO vendorVO) {

Map<String, Object> paramMap = new HashMap<String, Object>();

int vendor\_id = vendorVO.getId();

paramMap.put("vendor\_id", vendor\_id);

String[] categories = vendorVO.getCategories()

.substring(1, vendorVO.getCategories().length()).split(",");

int categories\_id[] = new int[categories.length];

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

if (!categories[i].equals("")) {

categories\_id[i] = Integer.parseInt(categories[i]);

}

}

paramMap.put("categories\_id", categories\_id);

this.serviceDistributionDao.deleteCategories(paramMap);

}

public List<CommunityVO> getCommunityList(int id) {

return this.serviceDistributionDao.getCommunityList(id);

}

public List<CategoriesVO> getCategoriesList(int id) {

return this.serviceDistributionDao.getCategoriesList(id);

}

public void updateUser(Integer user\_id, int vendor\_id) {

this.serviceDistributionDao.updateUser(user\_id, vendor\_id);

}

public List<CategoriesVO> getProductsCategoryList(int vendor\_id) {

return this.serviceDistributionDao.getProductsCategoryList(vendor\_id);

}

public List<ClassificationsVO> getAllCategoriesList(int groupId) {

return this.serviceDistributionDao.getAllCategoriesList(groupId);

}

public void insertFee(VendorVO vendorVO) {

this.serviceDistributionDao.insertFee(vendorVO);

}

public void removeUserVenRela(Integer id) {

this.serviceDistributionDao.removeUserVenRela(id);

}

}

package com.yawa.busi.service;

import java.util.List;

import java.util.Map;

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

import org.springframework.stereotype.Service;

import com.yawa.base.dao.GenericDAO;

import com.yawa.base.service.BaseService;

import com.yawa.busi.dao.SuggestionsDao;

import com.yawa.busi.model.SuggestionsCommVO;

import com.yawa.busi.model.SuggestionsVO;

import com.yawa.http.model.SuggestDetailModel;

import com.yawa.http.model.SuggestListModel;

import com.yawa.util.model.Page;

@Service

public class SuggestionsService extends BaseService<SuggestionsVO, Integer>{

@Autowired

private SuggestionsDao suggestionsDao;

@Override

public GenericDAO<SuggestionsVO, Integer> getGenericDao() {

return this.suggestionsDao;

}

/\*==============接口===============================\*/

public List<SuggestListModel> getSuggestPageList(int start, int limit,

String title, String vendor\_name,String state) {

return suggestionsDao.getSuggestPageList(start,limit,title.trim(),vendor\_name.trim(),state);

}

public SuggestDetailModel getSuggestDetailById(Integer suggest\_id) {

SuggestDetailModel detailModel = new SuggestDetailModel();

SuggestionsVO suggestVo = suggestionsDao.find(suggest\_id);

if(suggestVo != null){

detailModel.setId(suggestVo.getId());

detailModel.setTitle(suggestVo.getTitle());

detailModel.setContent(suggestVo.getContent());

detailModel.setVendor\_name(suggestVo.getVendor\_name());

detailModel.setCreate\_time(suggestVo.getCreate\_time());

detailModel.setComments(suggestionsDao.getCommentsList(suggestVo.getId()));

}

return detailModel;

}

public Page<SuggestionsCommVO> getCommentsList(Map<String,Object> paramMap, int startIdx, int limitIdx) {

long totalCount = suggestionsDao.getCommentsListCount(paramMap);

Page<SuggestionsCommVO> page = new Page<SuggestionsCommVO>(totalCount);

page.setRows(suggestionsDao.getCommentsList(paramMap, startIdx, limitIdx));

return page;

}

public void insertComment(SuggestionsCommVO commentVo){

suggestionsDao.insertComment(commentVo);

}

public void updateSuggestState(Integer suggest\_id,String state){

suggestionsDao.updateSuggestState(suggest\_id,state);

}

}

package com.yawa.core.controller;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.apache.commons.lang.StringUtils;

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.ResponseBody;

import com.yawa.core.model.User;

import com.yawa.core.service.UserService;

import com.yawa.util.MD5PasswordEncrptor;

import com.yawa.util.context.SecurityContextUtil;

import com.yawa.util.model.ResponseData;

@Controller

@RequestMapping("/loginManager")

public class LoginController {

@Autowired

private UserService userService;

/\*\*

\* 跳转到登录页面

\*/

@RequestMapping("/toLoginPage")

public String login() {

return "jsp/login/login";

}

/\*\*

\* 登录认证

\*/

@RequestMapping("/login")

@ResponseBody

public ResponseData login(String name,String password,HttpServletResponse response,HttpServletRequest request) {

User user = new User();

user.setLogin\_name(name);

User find = this.userService.findUser(user);

if(find == null){

return new ResponseData(true,"notFound");

}else{

if(!(find.getPassword\_digest().equals(MD5PasswordEncrptor.encodePassword(password)))){

return new ResponseData(true,"passError");

}

if(find.getState().equals("0")){

return new ResponseData(true,"wuxiao");

}

request.getSession().setAttribute(SecurityContextUtil.SPRING\_SESSION\_LOGIN\_USER\_KEY, find);

if(StringUtils.isEmpty(find.getVendor\_id())){//还没有完善商户资料

return new ResponseData(true,"addLife");

}else {

return new ResponseData(true,"distrMain");

}

}

}

/\*\*

\* 公益活动注册页面

\*/

@RequestMapping("/benefitReg")

public String benefitActivity() {

return "jsp/register/benefitReg";

}

/\*\*

\* 登录成功，还没有完善商户资料，跳转到加入慧生活展示页面

\*/

@RequestMapping("/toAddLifePage")

public String toAddLifePage() {

return "jsp/login/addLife";

}

/\*\*

\* 登录成功，已经完善商户资料，跳转到经营页面

\*/

@RequestMapping("/toDistrMainPage")

public String toDistrMainPage() {

return "jsp/distr/distrMain";

}

/\*\*

\* 登录退出，销毁session跳转到登录页面

\*/

@RequestMapping("/logout")

public String logout(HttpServletRequest request) {

request.getSession().invalidate();

// request.getSession().removeAttribute(SecurityContextUtil.SPRING\_SESSION\_LOGIN\_USER\_KEY);

// 移除用户，但session不变，下次登陆的时候看到的sessionID还是一样的

return "jsp/login/login";

}

/\*\*

\* 登录退出，销毁session跳转到首页页面

\*/

@RequestMapping("/toLoginOut")

public String toLoginOut(HttpServletRequest request) {

request.getSession().invalidate();

//request.getSession().removeAttribute(SecurityContextUtil.SPRING\_SESSION\_LOGIN\_USER\_KEY);

//移除用户，但session不变，下次登陆的时候看到的sessionID还是一样的

return "index";

}

/\*\*

\* 得到页面验证码

\*/

@RequestMapping("/getYanzhengCode")

@ResponseBody

public ResponseData getYanzhengCode(HttpServletResponse response,HttpServletRequest request) {

String code=(String)request.getSession().getAttribute("yanzhengCode");

return new ResponseData(true,code.toLowerCase());

}

/\*\*

\* 找回密码，跳转到重置密码页面

\*/

@RequestMapping("/toFindPwdStep3")

@ResponseBody

public ResponseData toFindPwdStep3(String loginName,HttpServletRequest request) {

request.getSession().setAttribute("loginName", loginName);

request.getSession().setAttribute("loginNameFlag", MD5PasswordEncrptor.encodePassword(loginName));

return ResponseData.SUCCESS\_NO\_DATA;

}

}

package com.yawa.core.controller;

import java.util.HashMap;

import java.util.Map;

import java.util.Random;

import javax.servlet.http.HttpServletRequest;

import org.apache.commons.lang.StringUtils;

import org.apache.commons.mail.EmailException;

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.ResponseBody;

import com.yawa.core.model.User;

import com.yawa.core.service.UserService;

import com.yawa.util.MD5PasswordEncrptor;

import com.yawa.util.context.SecurityContextUtil;

import com.yawa.util.email.MailSender;

import com.yawa.util.model.ResponseData;

import com.yawa.util.sms.HttpXmlUtil;

@Controller

@RequestMapping("/registerManager")

public class RegisterController {

@Autowired

private UserService userService;

/\*\*

\* 跳转到邮箱注册页面

\*/

@RequestMapping("/toRegisterEmailPage")

public String toRegisterEmailPage() {

return "jsp/register/registerEmail";

}

/\*\*

\* 跳转到手机注册页面

\*/

@RequestMapping("/toRegisterMobilePage")

public String toRegisterMobilePage() {

return "jsp/register/registerMobile";

}

/\*\*

\* 跳转到商户协议页面

\*/

@RequestMapping("/toMerchantProtocolPage")

public String toMerchantProtocolPage() {

return "jsp/register/merchantProtocol";

}

/\*\*

\* 邮箱是否已注册

\* @throws EmailException

\*/

@RequestMapping("/hasEmailExist")

@ResponseBody

public ResponseData hasEmailExist(String email){

User user=new User();

user.setEmail(email);

User find=this.userService.findUser(user);

if(find==null){

return new ResponseData(true,"n");

}else{

return new ResponseData(true,"y");

}

}

/\*\*

\* 发送邮箱验证码----暂时无用

\* @throws EmailException

\*/

@RequestMapping("/sendEmailCode")

@ResponseBody

public ResponseData sendEmailCode(String email,String code,HttpServletRequest request) throws EmailException {

MailSender ms=new MailSender();

Map<String,String> map=new HashMap<String,String>();

map.put("CheckCode", code);

ms.sendEmail("慧生活申请账号邮件验证", email, "Email\_CheckCodeTmpt", map);

request.getSession().setAttribute("emailOrMobileCode", code);

return ResponseData.SUCCESS\_NO\_DATA;

}

/\*\*

\* 邮箱用户注册

\*/

@RequestMapping("/emailRegister")

public String emailRegister(String email,String pwd,HttpServletRequest request) {

User user = new User();

user.setLogin\_name(email);

user.setEmail(email);

user.setPassword\_digest(MD5PasswordEncrptor.encodePassword(pwd));

userService.insertEntity(user);

User find = this.userService.findUser(user);

request.getSession().removeAttribute(SecurityContextUtil.SPRING\_SESSION\_LOGIN\_USER\_KEY);

request.getSession().setAttribute(SecurityContextUtil.SPRING\_SESSION\_LOGIN\_USER\_KEY, find);

return "jsp/login/addLife";

}

/\*\*

\* 手机是否已注册

\* @throws EmailException

\*/

@RequestMapping("/hasMobileExist")

@ResponseBody

public ResponseData hasMobileExist(String mobile){

User user=new User();

user.setMobile(mobile);

User find=this.userService.findUser(user);

if(find==null){

return new ResponseData(true,"n");

}else{

return new ResponseData(true,"y");

}

}

/\*\*

\* 发送手机验证码

\* @throws EmailException

\*/

@RequestMapping("/sendMobileCode")

@ResponseBody

public ResponseData sendMobileCode(String mobile,String code,HttpServletRequest request) throws EmailException {

//此处调用发送手机短信接口

HttpXmlUtil hxu=new HttpXmlUtil();

hxu.send(mobile,code);

request.getSession().setAttribute("emailOrMobileCode", code);

return ResponseData.SUCCESS\_NO\_DATA;

}

/\*\*

\* 手机用户注册

\*/

@RequestMapping("/mobileRegister")

public String mobileRegister(String mobile,String pwd,HttpServletRequest request) {

User user=new User();

user.setLogin\_name(mobile);

user.setMobile(mobile);

user.setPassword\_digest(MD5PasswordEncrptor.encodePassword(pwd));

userService.insertEntity(user);

User find = this.userService.findUser(user);

request.getSession().removeAttribute(SecurityContextUtil.SPRING\_SESSION\_LOGIN\_USER\_KEY);

request.getSession().setAttribute(SecurityContextUtil.SPRING\_SESSION\_LOGIN\_USER\_KEY, find);

return "jsp/login/addLife";

}

/\*\*

\* 跳转到商户资料完善页面

\*/

@RequestMapping("/toRegisterSeviceDistributionPage")

public String toRegisterSeviceDistributionPage(String groupId,HttpServletRequest request) {

request.getSession().setAttribute("groupId",groupId);

return "jsp/register/serviceDistribution";

}

/\*\*

\* 修改密码

\*/

@RequestMapping("/changePwd")

@ResponseBody

public ResponseData changePwd(String loginName,String pwd,String loginNameFlag,HttpServletRequest request) {

try{

User user=new User();

if(StringUtils.isNotEmpty(loginNameFlag)){//重置密码操作

if(MD5PasswordEncrptor.encodePassword(loginName).equals(loginNameFlag)){

user.setLogin\_name(loginName);

}else{

return new ResponseData(false,"invalid");

}

}else{

user.setLogin\_name(loginName);

}

user.setPassword\_digest(MD5PasswordEncrptor.encodePassword(pwd));

userService.updateEntity(user);

return new ResponseData(true,"y");

}catch(Exception e){

return new ResponseData(true,"n");

}

}

/\*\*

\* 产生并发送验证码

\* @param registerName 注册账号

\* @param type 注册类型 email-邮箱注册;mobile-手机注册

\* @param request

\* @return

\*/

@RequestMapping("/sendCode")

@ResponseBody

public ResponseData sendCode(String registerName,String type,String flg,HttpServletRequest request){

try {

String code="";

String title="";

Random random=new Random();

for(int i=0;i<6;i++){

code += random.nextInt(10);

}

request.getSession().setAttribute("emailOrMobileCode", code);

if(type.equals("email")){

MailSender ms=new MailSender();

Map<String,String> map=new HashMap<String,String>();

map.put("CheckCode", code);

if(flg.equals("1")){//邮箱注册

map.put("code1","申请账号邮件验证");

map.put("code2","邮件注册账号");

title = "慧生活申请账号邮件验证";

}else if(flg.equals("2")){//密码找回

map.put("code1","找回密码身份验证");

map.put("code2","邮件找回密码身份验证");

title = "慧生活找回密码身份验证";

}

ms.sendEmail(title, registerName, "Email\_CheckCodeTmpt", map);

}else if(type.equals("mobile")){

HttpXmlUtil hxu=new HttpXmlUtil();

hxu.send(registerName,code);

}

return new ResponseData(true,"y");

} catch (Exception e) {

e.printStackTrace();

return new ResponseData(true,"n");

}

}

/\*\*

\* 验证邮箱或手机验证码是否正确

\*/

@RequestMapping("/emailOrMobileCodeValid")

@ResponseBody

public ResponseData emailOrMobileCodeValid(String code,HttpServletRequest request){

String scode=(String)request.getSession().getAttribute("emailOrMobileCode");

if(scode.toLowerCase().equals(code.toLowerCase())){

return new ResponseData(true,"y");

}else{

return new ResponseData(true,"n");

}

}

}

package com.yawa.core.controller;

import org.apache.commons.mail.EmailException;

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.ResponseBody;

import com.yawa.core.model.User;

import com.yawa.core.service.UserService;

@Controller

@RequestMapping("/user")

public class UserController {

@Autowired

private UserService userService;

@RequestMapping("/index")

public String index() {

return "core/user";

}

/\*\*

\* 根据ID获取一个用户

\* @param id

\* @return

\* @throws EmailException

\*/

@RequestMapping(value = "/queryOne")

@ResponseBody

public User queryById(Integer id) throws EmailException {

return userService.findEntity(id);

}

}

package com.yawa.base.controller;

import java.io.File;

import java.io.IOException;

import java.util.HashMap;

import java.util.Map;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.apache.commons.lang.StringUtils;

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.ResponseBody;

import com.yawa.busi.util.ConstantItems;

import com.yawa.util.attach.Asset;

import com.yawa.util.attach.AssetHelper;

import com.yawa.util.file.FileDownUtils;

@Controller

@RequestMapping("/fileDownLoad")

public class FileLoadController {

@Autowired

private AssetHelper assetHelper;

/\*\*

\* 判断附件是否存在

\* @param id

\* @param response

\*/

@RequestMapping(value="/checkAttachs")

public void checkAttchs(int id,String type,HttpServletResponse response){

String message = "";

boolean success = false;

if(StringUtils.isEmpty(type)||id==0){

message = "id，类型不能为空!";

}else{

Asset attach = new Asset();

attach = this.assetHelper.getAssetByViewable(type, id);

if(attach == null){

message = "该条消息没有附件!";

}else if(StringUtils.isEmpty(attach.getAttachment\_store\_name())){

message = "附件名称不能为空!";

}else{

assetHelper.init();

if(!(assetHelper.getStoreFileDir(attach,ConstantItems.CONST\_MAP.get("ASSET\_ORIGIN")).exists())){

message = "该附件不存在!";

}else{

success = true;

}

}

}

response.setCharacterEncoding("UTF-8");

response.setContentType("text/html");

String json = "{\"success\":" + success + ",\"message\":\"" + message + "\"}";

try {

response.getWriter().print(json);

} catch (IOException e) {

e.printStackTrace();

}

}

/\*\*

\* 附件下载

\*

\* @param request

\* @param response

\* @throws Exception

\*/

@RequestMapping(value = "/download")

@ResponseBody

public void fileDownLoad(HttpServletRequest request,

HttpServletResponse response) {

String id = request.getParameter("id");

String type = request.getParameter("type");

Asset attach = new Asset();

attach = this.assetHelper.getAssetByViewable(type, Integer.parseInt(id));

assetHelper.init();

try {

File file = assetHelper.getStoreFileDir(attach, ConstantItems.CONST\_MAP.get("ASSET\_ORIGIN"));

FileDownUtils.downLoadFile(file, response);

} catch (IOException e) {

e.printStackTrace();

}

}

/\*\*

\* 下载微信手机版app

\* @param request

\* @param response

\* @return

\*/

@RequestMapping(value = "/downApp")

@ResponseBody

public Map<String, String> downApp(HttpServletRequest request,HttpServletResponse response){

Asset asset1 = new Asset();

Asset asset2 = new Asset();

asset1 = assetHelper.downApp("Android");

asset2 = assetHelper.downApp("IOS");

asset1.setViewable\_type(ConstantItems.CONST\_MAP.get("VIEABLE\_APP"));

asset2.setViewable\_type(ConstantItems.CONST\_MAP.get("VIEABLE\_APP"));

File file1 = assetHelper.getStoreFileDir(asset1, ConstantItems.CONST\_MAP.get("ASSET\_ORIGIN"));

File file2 = assetHelper.getStoreFileDir(asset2, ConstantItems.CONST\_MAP.get("ASSET\_ORIGIN"));

Map<String, String> params = new HashMap<String, String>();

params.put("isAndroidExists",file1.exists()==true?"1":"0");

params.put("isIOSExists",file2.exists()==true?"1":"0");

params.put("Android", ConstantItems.CONST\_MAP.get("ASSET\_INTERVIEW")+"/"+ConstantItems.CONST\_MAP.get("VIEABLE\_APP")+"/"+asset1.getId()+"/"+ConstantItems.CONST\_MAP.get("ASSET\_ORIGIN")+"/"+asset1.getAttachment\_store\_name());

params.put("IOS", ConstantItems.CONST\_MAP.get("ASSET\_INTERVIEW")+"/"+ConstantItems.CONST\_MAP.get("VIEABLE\_APP")+"/"+asset2.getId()+"/"+ConstantItems.CONST\_MAP.get("ASSET\_ORIGIN")+"/"+asset2.getAttachment\_store\_name());

return params;

}

}package com.yawa.base.service;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import org.apache.commons.mail.EmailException;

import org.apache.ibatis.session.SqlSession;

import com.yawa.base.model.NotifierVO;

import com.yawa.busi.util.ConstantItems;

import com.yawa.util.email.MailSender;

import com.yawa.util.sms.HttpXmlUtil;

public class NoticeService {

private SqlSession sqlSession;

public SqlSession getSqlSession() {

return sqlSession;

}

public void setSqlSession(SqlSession sqlSession) {

this.sqlSession = sqlSession;

}

public void noticeMessage(String busi\_type,String category,int user\_id,String content){

Map<String,Object> map=new HashMap<String,Object>();

map.put("category", category);

map.put("user\_id", user\_id);

List<NotifierVO> list = sqlSession.selectList("com.yawa.base.model.NotifierVO.getList",map);

for(int i=0;i<list.size();i++){

NotifierVO n = list.get(i);

n.setBusi\_type(busi\_type);

if(n.getSend\_method().equals("1")||n.getSend\_method().equals("3")){

HttpXmlUtil hxu=new HttpXmlUtil();

hxu.sendSms(n.getMobile(), content);

n.setSend\_to(n.getMobile());

n.setSend\_content(content);

sqlSession.insert("com.yawa.base.model.NotifierVO.insertLog",n);

}

if(n.getSend\_method().equals("2")||n.getSend\_method().equals("3")){

MailSender ms=new MailSender();

Map<String,String> mapM=new HashMap<String,String>();

//map.put("CheckCode", code);

try {

ms.sendEmail("最新通知", n.getEmail(), "Email\_MessageYpt", mapM);

} catch (EmailException e) {

e.printStackTrace();

}

n.setSend\_content(ConstantItems.CONST\_MAP.get("Email\_MessageYpt"));

n.setSend\_to(n.getEmail());

sqlSession.insert("com.yawa.base.model.NotifierVO.insertLog",n);

}

}

}

}

package com.yawa.util;

import java.util.Map;

import org.springframework.context.ApplicationContext;

import org.springframework.context.ApplicationContextAware;

/\*\*

\* Spring应用上下文持有类

\* @author 虞贵涛

\* @create at 2012-05-16

\* @version 1.0

\*/

public class ApplicationContextHolder implements ApplicationContextAware {

private static ApplicationContext context;

public void setApplicationContext(ApplicationContext applicationContext) {

context = applicationContext;

}

public static void setAppContext(ApplicationContext applicationContext) {

context = applicationContext;

}

public static ApplicationContext getApplicationContext() {

return context;

}

public static Object getBean(String name) {

return getApplicationContext().getBean(name);

}

public static <T> T getBean(Class<T> clazz) {

return getApplicationContext().getBean(clazz);

}

public static <T> Map<String, T> getBeansOfType(Class<T> clazz) {

return getApplicationContext().getBeansOfType(clazz);

}

}

package com.yawa.util;

import java.lang.annotation.Annotation;

import java.lang.reflect.Field;

import java.lang.reflect.Method;

import java.util.HashMap;

import java.util.Map;

import org.apache.commons.lang.StringUtils;

public class BeanUtils {

/\*\*

\* 根据注解类获取含有该注解的成员变量的名字，如果有多个只取第一个。没有的话返回null

\* @param annClass

\* @param objClass

\* @return

\*/

public static Field getAnnotFieldByAnnotClass(Class<? extends Annotation> annClass,Class<?> objClass){

Field[] fields = objClass.getDeclaredFields();

for(Field field : fields){

if(field.isAnnotationPresent(annClass)){

return field;

}

}

return null;

}

/\*\*

\* 返回方法数组（methods）中方法名为name的方法

\*

\* @param methods

\* @param name

\* @return

\*/

private static Method getMethodByName(Method[] methods, String name) {

for (Method method : methods) {

if (method.getName().equals(name)) {

return method;

}

}

return null;

}

/\*\*

\* 获取方法名称

\* @param prefix

\* @param fieldName

\* @return

\*/

public static String getMethodName(String prefix,String fieldName){

if(StringUtils.isEmpty(fieldName)){

return "";

}

return prefix + Character.toUpperCase(fieldName.charAt(0)) + fieldName.substring(1);

}

/\*\*

\* 将bean对象转换成Map键值对的形式

\* @param bean

\* @return

\*/

public static Map<String,Object> convert2Map(Object obj){

Map<String, Object> retMap = new HashMap<String, Object>();

if(obj == null){

return retMap;

}

// 获得对象的所有属性

Class<? extends Object> clazz = obj.getClass();

Field[] fields = clazz.getDeclaredFields();

Method[] methods = clazz.getMethods();

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

// 获取数组中对应的属性

Field field = fields[i];

String fieldName = field.getName();

// 获得相应属性的getXXX和setXXX方法名称

String getName = "get" + Character.toUpperCase(fieldName.charAt(0)) + fieldName.substring(1);

// 获取相应的方法

try {

Method getMethod = getMethodByName(methods,getName);

if (getMethod != null) {

Object value = getMethod.invoke(obj);

if(value instanceof String){

value = StringUtils.trimToNull((String)value);

}

retMap.put(fieldName, value);

}

} catch (Exception e) {

e.printStackTrace();

}

}

return retMap;

}

}

package com.yawa.util;

public class ToolKit{

public static String ToText(String text) {

return text.replaceAll("&amp;","&").replaceAll("&nbsp;"," ").replaceAll("&gt;",">").replaceAll("&lt;","<").replaceAll("<br>","\r\n");

}

public static String toHtml(String html) {

return html.replaceAll("&","&amp;").replaceAll(" ","&nbsp;").replaceAll(">","&gt;").replaceAll("<","&lt;").replaceAll("\r\n","<br />");

}

/\*\*

\* 因為JSON傳中的特殊字符，無法解析問題，特與OA辦公平台約定特殊字符加密方式

\* @param replacement

\* @return

\*/

public static String findReplaceChar(String replacement){

return replacement.replace("&quto;", "\"").replace("&ctrl;","\r\n").replace("@mh;",":");

}

public static String replaceChar(String replacement){

return replacement.replace("\"","&quot;").replace("\r\n","&ctrl;").replace( ":","@mh;");

}

}

package com.yawa.util.attach;

import java.io.File;

/\*\*

\* 附件上传公共类

\*

\* @author wrz

\* @since 2014-09-11

\*/

public class Asset {

// 主键自动生成

private int id;

// 主表ID

private int viewable\_id;

// 主表类型

private String viewable\_type;

// 附件宽度

private int attachment\_width;

// 附件高度

private int attachment\_height;

// 附件大小

private int attachment\_file\_size;

// 附件类型

private String attachment\_content\_type = "";

// 文件上传名称

private String attachment\_file\_name = "";

// 文件存储名称

private String attachment\_store\_name = "";

// 提示文字

private String alt = "";

// 文件的类型 IMAGE FILE等

private String type = "";

// 文件的顺序

private int position;

// 小尺寸

private String mini\_size = "";

// 中等尺寸

private String medium\_size = "";

// 图片切割位置

private String imageCutPos = "";

// 上传的文件

private File uploadFile = null;

/\*\*

\* @return the id

\*/

public int getId() {

return id;

}

/\*\*

\* @param id

\* the id to set

\*/

public void setId(int id) {

this.id = id;

}

/\*\*

\* @return the viewable\_id

\*/

public int getViewable\_id() {

return viewable\_id;

}

/\*\*

\* @param viewable\_id

\* the viewable\_id to set

\*/

public void setViewable\_id(int viewable\_id) {

this.viewable\_id = viewable\_id;

}

/\*\*

\* @return the viewable\_type

\*/

public String getViewable\_type() {

return viewable\_type;

}

/\*\*

\* @param viewable\_type

\* the viewable\_type to set

\*/

public void setViewable\_type(String viewable\_type) {

this.viewable\_type = viewable\_type;

}

/\*\*

\* @return the attachment\_width

\*/

public int getAttachment\_width() {

return attachment\_width;

}

/\*\*

\* @param attachment\_width

\* the attachment\_width to set

\*/

public void setAttachment\_width(int attachment\_width) {

this.attachment\_width = attachment\_width;

}

/\*\*

\* @return the attachment\_height

\*/

public int getAttachment\_height() {

return attachment\_height;

}

/\*\*

\* @param attachment\_height

\* the attachment\_height to set

\*/

public void setAttachment\_height(int attachment\_height) {

this.attachment\_height = attachment\_height;

}

/\*\*

\* @return the attachment\_file\_size

\*/

public int getAttachment\_file\_size() {

return attachment\_file\_size;

}

/\*\*

\* @param attachment\_file\_size

\* the attachment\_file\_size to set

\*/

public void setAttachment\_file\_size(int attachment\_file\_size) {

this.attachment\_file\_size = attachment\_file\_size;

}

/\*\*

\* @return the attachment\_content\_type

\*/

public String getAttachment\_content\_type() {

return attachment\_content\_type;

}

/\*\*

\* @param attachment\_content\_type

\* the attachment\_content\_type to set

\*/

public void setAttachment\_content\_type(String attachment\_content\_type) {

this.attachment\_content\_type = attachment\_content\_type;

}

/\*\*

\* @return the attachment\_file\_name

\*/

public String getAttachment\_file\_name() {

return attachment\_file\_name;

}

/\*\*

\* @param attachment\_file\_name

\* the attachment\_file\_name to set

\*/

public void setAttachment\_file\_name(String attachment\_file\_name) {

this.attachment\_file\_name = attachment\_file\_name;

}

/\*\*

\* @return the attachment\_store\_name

\*/

public String getAttachment\_store\_name() {

return attachment\_store\_name;

}

/\*\*

\* @param attachment\_store\_name

\* the attachment\_store\_name to set

\*/

public void setAttachment\_store\_name(String attachment\_store\_name) {

this.attachment\_store\_name = attachment\_store\_name;

}

/\*\*

\* @return the alt

\*/

public String getAlt() {

return alt;

}

/\*\*

\* @param alt

\* the alt to set

\*/

public void setAlt(String alt) {

this.alt = alt;

}

/\*\*

\* @return the type

\*/

public String getType() {

return type;

}

/\*\*

\* @param type

\* the type to set

\*/

public void setType(String type) {

this.type = type;

}

/\*\*

\* @return the position

\*/

public int getPosition() {

return position;

}

/\*\*

\* @param position

\* the position to set

\*/

public void setPosition(int position) {

this.position = position;

}

/\*\*

\* @return the mini\_size

\*/

public String getMini\_size() {

return mini\_size;

}

/\*\*

\* @param mini\_size

\* the mini\_size to set

\*/

public void setMini\_size(String mini\_size) {

this.mini\_size = mini\_size;

}

/\*\*

\* @return the medium\_size

\*/

public String getMedium\_size() {

return medium\_size;

}

/\*\*

\* @param medium\_size

\* the medium\_size to set

\*/

public void setMedium\_size(String medium\_size) {

this.medium\_size = medium\_size;

}

/\*\*

\* @return the imageCutPos

\*/

public String getImageCutPos() {

return imageCutPos;

}

/\*\*

\* @param imageCutPos the imageCutPos to set

\*/

public void setImageCutPos(String imageCutPos) {

this.imageCutPos = imageCutPos;

}

/\*\*

\* @return the uploadFile

\*/

public File getUploadFile() {

return uploadFile;

}

/\*\*

\* @param uploadFile

\* the uploadFile to set

\*/

public void setUploadFile(File uploadFile) {

this.uploadFile = uploadFile;

}

}

package com.yawa.util.email;

import java.util.Iterator;

import java.util.Map;

import java.util.Set;

import org.apache.commons.mail.DefaultAuthenticator;

import org.apache.commons.mail.Email;

import org.apache.commons.mail.EmailException;

import org.apache.commons.mail.HtmlEmail;

import com.yawa.busi.util.ConstantItems;

/\*\*

\* 发送邮件

\* @author wrz

\*/

public class MailSender{

public void sendEmail(String title,String to\_addr,String email\_tmpt,Map<String,String> params) throws EmailException{

new sentEmailThread(title,to\_addr,email\_tmpt,params).start();

}

private class sentEmailThread extends Thread {

private String to\_addr = "";

private String email\_tmpt = "";

private String title = "";

private Map<String,String> params = null;

public sentEmailThread(String title,String to\_addr,String email\_tmpt,Map<String,String> params){

this.title = title;

this.to\_addr = to\_addr;

this.email\_tmpt = email\_tmpt;

this.params = params;

}

@Override

public void run() {

Email email = new HtmlEmail();

email.setHostName(ConstantItems.CONST\_MAP.get("Email\_Host"));

email.setSmtpPort(Integer.parseInt(ConstantItems.CONST\_MAP.get("Email\_Port")));

email.setAuthenticator(new DefaultAuthenticator(

ConstantItems.CONST\_MAP.get("Email\_User"),

ConstantItems.CONST\_MAP.get("Email\_Pass")));

email.setSSLOnConnect(true);

try{

email.setFrom(ConstantItems.CONST\_MAP.get("Email\_From"),"惠生活服务");

email.setSubject(title);

String emailTmpt = ConstantItems.CONST\_MAP.get(email\_tmpt);

// 替换邮件模板中的变量

Set<String> key = params.keySet();

for (Iterator<String> it = key.iterator(); it.hasNext();) {

String s = it.next();

emailTmpt = emailTmpt.replaceAll(s, params.get(s));

}

email.setMsg(emailTmpt);

email.addTo(to\_addr);

email.send();

}catch(EmailException ex){

ex.printStackTrace();

}

}

}

}

package com.yawa.util.filter;

import java.io.IOException;

import java.util.Enumeration;

import java.util.Map;

import javax.servlet.Filter;

import javax.servlet.FilterChain;

import javax.servlet.FilterConfig;

import javax.servlet.ServletContext;

import javax.servlet.ServletException;

import javax.servlet.ServletRequest;

import javax.servlet.ServletResponse;

import org.apache.commons.logging.Log;

import org.apache.commons.logging.LogFactory;

import com.yawa.busi.util.ConstantItems;

import com.yawa.util.PropertiesReader;

public class ContextParamFilter

implements Filter {

private static final Log log = LogFactory.getLog(ContextParamFilter.class);

public void init(FilterConfig config)

throws ServletException {

ServletContext application = config.getServletContext();

PropertiesReader reader = PropertiesReader.getInstance();

try {

// 初始化config.properties 配置文件

Map<String,String> constMap = ConstantItems.CONST\_MAP;

constMap.clear();

Enumeration<?> enu = reader.getPropertNames();

while (enu.hasMoreElements()) {

String key = (String)enu.nextElement();

constMap.put(key, reader.getValue(key).trim());

}

ConstantItems.YYC\_INTERFACE\_PATH = constMap.get("YYC\_PATH");

ConstantItems.WEIXIN\_ACCESS\_PATH = constMap.get("WEIXIN\_PATH");

ConstantItems.WEIXIN\_IMAGE\_PATH = constMap.get("ASSET\_INTERVIEW");

application.setAttribute("CONST\_MAP", constMap);

} catch (Exception e1) {

log.info(e1.getMessage());

e1.printStackTrace();

}

}

@Override

public void destroy() {

}

@Override

public void doFilter(ServletRequest arg0, ServletResponse arg1,

FilterChain arg2) throws IOException, ServletException {

}

}

package com.yawa.util.filter;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.FilterChain;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import org.apache.commons.logging.Log;

import org.apache.commons.logging.LogFactory;

import org.springframework.web.filter.OncePerRequestFilter;

import com.yawa.core.model.User;

import com.yawa.util.context.SecurityContextHolder;

/\*\*

\* @Title:登录过滤

\* @Description:

\* @Author:yuzhe

\* @Since:2013-4-23

\* @Version:1.1.0

\*/

public class SessionFilter extends OncePerRequestFilter {

static Log logger = LogFactory.getLog(SessionFilter.class);

@Override

protected void doFilterInternal(HttpServletRequest request,

HttpServletResponse response, FilterChain filterChain)

throws ServletException, IOException {

SecurityContextHolder.setHttpRequest(request);

SecurityContextHolder.setSession(request.getSession());

// 不过滤的uri

String[] notFilter = new String[] {

"index.jsp",

"findPwdStep1.jsp",

"findPwdStep2.jsp",

"findPwdStep3.jsp",

"yanzheng.jsp",

"findPwdSuccess.jsp"};

// 请求的uri

String uri = request.getRequestURI();

// 是否过滤

boolean doFilter = true;

for (String s : notFilter) {

if (uri.indexOf(s) != -1) {

// 如果uri中包含不过滤的uri，则不进行过滤

doFilter = false;

break;

}

}

if (!doFilter) {

// 如果不执行过滤，则继续

filterChain.doFilter(request, response);

} else {

// 执行过滤

// 从session中获取登录者实体

try{

User user = (User)request.getSession().getAttribute("currentUser");

if(user == null){

throw new RuntimeException("session过期");

}

filterChain.doFilter(request, response);

}catch(RuntimeException ex){

String loginPage = "";

int port = request.getServerPort();

if (port == 80) {

loginPage = "http://" + request.getServerName()

+ request.getContextPath() + "/index.jsp";

} else {

loginPage = "http://" + request.getServerName() + ":"

+ port + request.getContextPath() + "/index.jsp";;

}

// 如果session中不存在登录者实体，则弹出框提示重新登录

// 设置request和response的字符集，防止乱码

response.setCharacterEncoding("UTF-8");

PrintWriter out = response.getWriter();

StringBuilder builder = new StringBuilder();

builder.append("<script type=\"text/javascript\">");

builder.append("window.top.location.href='");

builder.append(loginPage);

builder.append("';");

builder.append("</script>");

out.print(builder.toString());

}

}

SecurityContextHolder.removeHttpRequest();

SecurityContextHolder.removeHttpSession();

}

}

package com.yawa.app.modal;

import java.util.Date;

import org.apache.commons.lang.time.DateFormatUtils;

import com.yawa.busi.util.ConstantItems;

import com.yawa.core.model.User;

import com.yawa.util.attach.Asset;

import com.yawa.util.context.SecurityContextUtil;

public class AppUploadImage {

private long images\_id;

private String images\_name = "";

private String images\_url = "";

private String images\_user\_account = "";

private String images\_user\_name = "";

private String images\_date = System.currentTimeMillis() + "";

private String images\_enable = "1";

public AppUploadImage(Asset asset,String imgType){

User user = SecurityContextUtil.getCurrentUser();

if(user != null){

this.images\_user\_account = user.getLogin\_name();

this.images\_user\_name = user.getLogin\_name();

}

if(asset != null){

this.images\_name = asset.getAttachment\_store\_name();

this.images\_url = imgType + "/images/" + DateFormatUtils.format(new Date(), "yyyy-MM-dd");

if(asset.getViewable\_type().equals(ConstantItems.CONST\_MAP.get("VIEWABLE\_HEAD\_REG"))){

// 商超头像

this.images\_id = asset.getViewable\_id() + 1000000000l;

}else if(asset.getViewable\_type().equals(ConstantItems.CONST\_MAP.get("PRODUCT\_IMG\_REG"))){

// 产品图片

this.images\_id = asset.getViewable\_id() + 5000000000l;

}

}

}

public long getImages\_id() {

return images\_id;

}

public void setImages\_id(long images\_id) {

this.images\_id = images\_id;

}

public String getImages\_name() {

return images\_name;

}

public void setImages\_name(String images\_name) {

this.images\_name = images\_name;

}

public String getImages\_url() {

return images\_url;

}

public void setImages\_url(String images\_url) {

this.images\_url = images\_url;

}

public String getImages\_user\_account() {

return images\_user\_account;

}

public void setImages\_user\_account(String images\_user\_account) {

this.images\_user\_account = images\_user\_account;

}

public String getImages\_user\_name() {

return images\_user\_name;

}

public void setImages\_user\_name(String images\_user\_name) {

this.images\_user\_name = images\_user\_name;

}

public String getImages\_date() {

return images\_date;

}

public void setImages\_date(String images\_date) {

this.images\_date = images\_date;

}

public String getImages\_enable() {

return images\_enable;

}

public void setImages\_enable(String images\_enable) {

this.images\_enable = images\_enable;

}

}

/\*\*

\* @Copyright:Copyright (c) 2011 - 2100

\* @Company:YHYH

\*/

package com.yawa.http.common;

import java.io.File;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Hashtable;

import java.util.List;

import java.util.Map;

import com.yawa.util.attach.Asset;

import net.sf.json.JSONObject;

/\*\*

\* @Title:数据传送对象基础类

\* @Description:数据传送对象基础类，将从webserice客户端接收到的参数组装为

\* @Author:wwl

\* @Since:2012-4-18

\* @Version:1.1.0

\*/

public class DataPkg {

// 命令字

private String strIntCmd;

// 返回码

private String retcode;

// 返回信息

private String retmsg;

// 入参vector

private JSONObject inParam;

// 返回列表

private Object outParam;

private String operId = "";

private String operName = "";

// 分页使用

private String total = "";

private Hashtable<String,Object> hashVO = null;

// 文件列表

private List<Asset> assets = null;;

// 文件附件

private File attachFile;

public DataPkg() {

retcode = ConstantItems.CODE\_COMMON\_SUCCESS;

retmsg = "数据处理成功";

strIntCmd = "";

operId = "";

inParam = new JSONObject();

outParam = new Object();

assets = new ArrayList<Asset>();

hashVO = new Hashtable<String,Object>();

attachFile = null;

total = "";

}

public String getStrIntCmd() {

return strIntCmd;

}

public void setStrIntCmd(String strIntCmd) {

this.strIntCmd = strIntCmd;

}

public String getRetcode() {

return retcode;

}

public void setRetcode(String retcode) {

this.retcode = retcode;

}

public String getRetmsg() {

return retmsg;

}

public void setRetmsg(String retmsg) {

this.retmsg = retmsg;

}

public JSONObject getInParam() {

return inParam;

}

public void setInParam(JSONObject inParam) {

this.inParam = inParam;

}

public Object getOutParam() {

return outParam;

}

public void setOutParam(Object outParam) {

this.outParam = outParam;

}

public List<Asset> getAssets() {

return assets;

}

public void setAssets(List<Asset> assets) {

this.assets = assets;

}

public File getAttachFile() {

return attachFile;

}

public void setAttachFile(File attachFile) {

this.attachFile = attachFile;

}

public String getOperId() {

return operId;

}

public void setOperId(String operId) {

this.operId = operId;

}

public String getOperName() {

return operName;

}

public void setOperName(String operName) {

this.operName = operName;

}

public Hashtable<String, Object> getHashVO() {

return hashVO;

}

public void setHashVO(Hashtable<String, Object> hashVO) {

this.hashVO = hashVO;

}

public String getTotal() {

return total;

}

public void setTotal(String total) {

this.total = total;

}

}

package com.yawa.http.common;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import org.apache.commons.httpclient.HttpClient;

import org.apache.commons.httpclient.HttpStatus;

import org.apache.commons.httpclient.NameValuePair;

import org.apache.commons.httpclient.methods.GetMethod;

import org.apache.commons.httpclient.methods.PostMethod;

import org.apache.commons.httpclient.params.HttpMethodParams;

public class HttpClientUtil {

public static String post(String URLString,NameValuePair[] pairs) {

StringBuffer response = new StringBuffer();

try {

HttpClient client = new HttpClient();

PostMethod method = new PostMethod(URLString);

method.addParameters(pairs);

method.getParams().setParameter(HttpMethodParams.HTTP\_CONTENT\_CHARSET, "UTF-8");

// 请求参数

try {

client.executeMethod(method); // 这一步就把文件上传了

if (method.getStatusCode() == HttpStatus.SC\_OK) {

BufferedReader reader = new BufferedReader(new InputStreamReader(method.getResponseBodyAsStream(), "UTF-8"));

String line = "";

while ((line = reader.readLine()) != null) {

response.append(line);

}

reader.close();

}

} catch (IOException e) {

System.out.println("执行HTTP Post请求" + URLString + "时，发生异常！");

e.printStackTrace();

} finally {

method.releaseConnection();

}

} catch (Exception e) {

e.printStackTrace();

}

return response.toString();

}

public static String get(String URLString) {

StringBuffer response = new StringBuffer();

try {

HttpClient client = new HttpClient();

GetMethod getMethod = new GetMethod(URLString);

getMethod.getParams().setParameter(HttpMethodParams.HTTP\_CONTENT\_CHARSET, "UTF-8");

// 请求参数

try {

client.executeMethod(getMethod); // 这一步就把文件上传了

if (getMethod.getStatusCode() == HttpStatus.SC\_OK) {

BufferedReader reader = new BufferedReader(new InputStreamReader(getMethod.getResponseBodyAsStream(), "UTF-8"));

String line = "";

while ((line = reader.readLine()) != null) {

response.append(line);

}

reader.close();

}

} catch (IOException e) {

System.out.println("执行HTTP get请求" + URLString + "时，发生异常！");

e.printStackTrace();

} finally {

getMethod.releaseConnection();

}

} catch (Exception e) {

e.printStackTrace();

}

return response.toString();

}

}

package com.yawa.http.common;

import java.util.ArrayList;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import com.yawa.util.attach.Asset;

public class RequestModel {

private List<Asset> assets = new ArrayList<Asset>();

private Map<String, Object> paraMap = new HashMap<String, Object>();

public List<Asset> getAssets() {

return assets;

}

public void setAssets(List<Asset> assets) {

this.assets = assets;

}

public Map<String, Object> getParaMap() {

return paraMap;

}

public void setParaMap(Map<String, Object> paraMap) {

this.paraMap = paraMap;

}

public void addAsset(Asset asset) {

this.getAssets().add(asset);

}

public void addParam(String key,String value) {

this.getParaMap().put(key, value);

}

public Object getParam(String key) {

return this.getParaMap().get(key);

}

}

package com.yawa.http.service;

import java.util.HashMap;

import java.util.Hashtable;

import java.util.List;

import java.util.Map;

import java.util.Random;

import net.sf.json.JSONObject;

import org.apache.commons.lang.StringUtils;

import org.apache.commons.mail.EmailException;

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

import org.springframework.stereotype.Service;

import org.apache.commons.logging.Log;

import org.apache.commons.logging.LogFactory;

import com.yawa.busi.model.ClassificationsVO;

import com.yawa.busi.model.CommunityVO;

import com.yawa.busi.model.VendorVO;

import com.yawa.busi.service.ServiceDistributionService;

import com.yawa.core.model.User;

import com.yawa.core.service.UserService;

import com.yawa.http.common.ConstantItems;

import com.yawa.http.common.DataPkg;

import com.yawa.http.model.VerifyCode;

import com.yawa.util.MD5PasswordEncrptor;

import com.yawa.util.attach.Asset;

import com.yawa.util.attach.AssetHelper;

import com.yawa.util.email.MailSender;

import com.yawa.util.sms.HttpXmlUtil;

@Service

public class AccountService {

@Autowired

private UserService userService;

@Autowired

private VerifyCodeService codeService;

@Autowired

private ServiceDistributionService vendorService;

@Autowired

private AssetHelper assetHelper;

public static Integer CODE\_MINUTES = 3;

private String getRandomCode(){

Random rand = new Random();

String result = "";

for(int i=0;i<6;i++){

result += rand.nextInt(10);

}

return result;

}

/\*\*

\* 发送邮箱验证码

\* @param datapkg

\* @return

\*/

public DataPkg sendEmailMsg(DataPkg datapkg) {

DataPkg rtdatapkg = new DataPkg();

JSONObject param = (JSONObject) datapkg.getInParam();

String email = "";

try { email = param.getString("email"); } catch (Exception e) {}

if(StringUtils.isEmpty(email)){

rtdatapkg.setRetcode(ConstantItems.CODE\_COMMON\_QOFAILED);

rtdatapkg.setRetmsg("注册邮箱不能为空！");

return rtdatapkg;

}

Map<String,String> paraMap = new HashMap<String,String>();

String randomCode = getRandomCode();

paraMap.put("CheckCode",randomCode);

paraMap.put("code1","申请账号邮件验证");

paraMap.put("code2","邮件注册账号");

try {

VerifyCode code = new VerifyCode();

code.setEmail(email);

code.setCode(randomCode);

code.setMinutes(CODE\_MINUTES);

codeService.insertEntity(code);

new MailSender().sendEmail("慧生活申请账号邮件验证", email, "Email\_CheckCodeTmpt", paraMap);

} catch (EmailException e) {

rtdatapkg.setRetcode(ConstantItems.CODE\_COMMON\_DBFAILED);

rtdatapkg.setRetmsg("邮箱验证码发送失败: " + e.getMessage());

return rtdatapkg;

}

rtdatapkg.setRetcode(ConstantItems.CODE\_COMMON\_SUCCESS);

rtdatapkg.setRetmsg("邮箱验证码发送成功！");

return rtdatapkg;

}

/\*\*

\* 发送手机验证码

\* @param datapkg

\* @return

\*/

public DataPkg sendMobleMsg(DataPkg datapkg) {

DataPkg rtdatapkg = new DataPkg();

JSONObject param = (JSONObject) datapkg.getInParam();

String mobile = "";

try { mobile = param.getString("mobile"); } catch (Exception e) {}

if(StringUtils.isEmpty(mobile)){

rtdatapkg.setRetcode(ConstantItems.CODE\_COMMON\_QOFAILED);

rtdatapkg.setRetmsg("注册手机号不能为空！");

return rtdatapkg;

}

String randomCode = getRandomCode();

try{

VerifyCode code = new VerifyCode();

code.setPhone(mobile);

code.setCode(randomCode);

code.setMinutes(CODE\_MINUTES);

codeService.insertEntity(code);

new HttpXmlUtil().send(mobile, randomCode);

}catch(Exception e){

rtdatapkg.setRetcode(ConstantItems.CODE\_COMMON\_DBFAILED);

rtdatapkg.setRetmsg("短信验证码发送失败: " + e.getMessage());

return rtdatapkg;

}

Hashtable<String, Object> ht = new Hashtable<String, Object>();

ht.put("code", randomCode);

rtdatapkg.setOutParam(ht);

rtdatapkg.setRetcode(ConstantItems.CODE\_COMMON\_SUCCESS);

rtdatapkg.setRetmsg("短信验证码发送成功！");

return rtdatapkg;

}

/\*\*

\* 账户注册

\* @param datapkg

\* @return

\*/

public DataPkg accountReg(DataPkg datapkg) {

DataPkg rtdatapkg = new DataPkg();

JSONObject param = (JSONObject) datapkg.getInParam();

User user = new User();

String code = "";

String password = "";

try {user.setLogin\_name(param.getString("login")); } catch (Exception e) {}

if(StringUtils.isEmpty(user.getLogin\_name())){

rtdatapkg.setRetcode(ConstantItems.CODE\_COMMON\_QOFAILED);

rtdatapkg.setRetmsg("注册账号不能为空！");

return rtdatapkg;

}

---END----