package com.nork.pano.model.scene;

import com.nork.common.util.Utils;

import com.nork.system.model.bo.SysDictionaryBo;

import java.io.Serializable;

import java.util.List;

/\*\*

\* Created by Administrator on 2017/7/11.

\*/

public class PanoramaVo implements Serializable{

/\*\*

\*

\*/

private static final long serialVersionUID = 1L;

/\*\* 设计方案ID \*\*/

private Integer planId;

/\*\* 三度logo \*\*/

private String logoPath;

/\*\* 渲染用户ID \*\*/

private Integer userId;

/\*\* 渲染用户昵称 \*\*/

private String userName;

/\*\* 渲染用户类型 \*\*/

private Integer userType;

/\*\* 公司名称 \*\*/

private String companyName;

/\*\* 公司LOGO \*\*/

private String companyLogoPath;

/\*\* 房型类型 \*\*/

private String houseTypeName;

/\*\* 标题 \*\*/

private String title;

/\*\* 描述 \*\*/

private String desc = Utils.getPropertyName("config/share","wx.share.desc","我在三度云享家分享了一款装修设计，点击查看全景漫游");

/\*\* 缩略图url \*\*/

private String thumbUrl;

/\*\* 窗口占比 \*\*/

private String windowsPercent;

/\*\*

\* 副本id

\*/

private Integer designPlanRenderSceneId;

/\*\*

\* 推荐方案id

\*/

private Integer designPlanRecommendedId;

/\*\*

\* 自动渲染得到的方案的id

\* @return

\*/

private Integer oneKeyDesignPlanId;

/\*\* 当前场景图id \*\*/

private Integer renderPicId;

/\*\* 异业联盟品牌ID集合\*/

private List<Integer> brandList;

//平台id

private Integer platformId;

//公司id

private Integer companyId;

//可见产品范围大小类

private List<SysDictionaryBo> sysDictionaryBOList;

//本公司品牌id集合

private List<Integer> ownBrandIdList;

//是否三度公司 0:不是, 1:是

private Integer isSandu = 0;

private String platformCode;

private String platformBussinessType;//平台业务类型

// 720方案来源(1:方案模块，2:店铺方案（可能方案对应几个店铺，需要传具体的店铺id），3:我的方案)

private Integer planSource;

// 店铺id(店铺方案的时候需使用店铺id查询店铺相关信息)

private Integer shopId;

public Integer getPlanSource() {

return planSource;

}

public void setPlanSource(Integer planSource) {

this.planSource = planSource;

}

public Integer getShopId() {

return shopId;

}

public void setShopId(Integer shopId) {

this.shopId = shopId;

}

public List<Integer> getBrandList() {

return brandList;

}

public void setBrandList(List<Integer> brandList) {

this.brandList = brandList;

}

public Integer getPlatformId() {

return platformId;

}

public void setPlatformId(Integer platformId) {

this.platformId = platformId;

}

public Integer getCompanyId() {

return companyId;

}

public void setCompanyId(Integer companyId) {

this.companyId = companyId;

}

public List<SysDictionaryBo> getSysDictionaryBOList() {

return sysDictionaryBOList;

}

public void setSysDictionaryBOList(List<SysDictionaryBo> sysDictionaryBOList) {

this.sysDictionaryBOList = sysDictionaryBOList;

}

public List<Integer> getOwnBrandIdList() {

return ownBrandIdList;

}

public void setOwnBrandIdList(List<Integer> ownBrandIdList) {

this.ownBrandIdList = ownBrandIdList;

}

public Integer getIsSandu() {

return isSandu;

}

public void setIsSandu(Integer isSandu) {

this.isSandu = isSandu;

}

public String getPlatformCode() {

return platformCode;

}

public void setPlatformCode(String platformCode) {

this.platformCode = platformCode;

}

public String getPlatformBussinessType() {

return platformBussinessType;

}

public void setPlatformBussinessType(String platformBussinessType) {

this.platformBussinessType = platformBussinessType;

}

public Integer getRenderPicId() {

return renderPicId;

}

public void setRenderPicId(Integer renderPicId) {

this.renderPicId = renderPicId;

}

public Integer getOneKeyDesignPlanId() {

return oneKeyDesignPlanId;

}

public void setOneKeyDesignPlanId(Integer oneKeyDesignPlanId) {

this.oneKeyDesignPlanId = oneKeyDesignPlanId;

}

public Integer getDesignPlanRenderSceneId() {

return designPlanRenderSceneId;

}

public void setDesignPlanRenderSceneId(Integer designPlanRenderSceneId) {

this.designPlanRenderSceneId = designPlanRenderSceneId;

}

public Integer getDesignPlanRecommendedId() {

return designPlanRecommendedId;

}

public void setDesignPlanRecommendedId(Integer designPlanRecommendedId) {

this.designPlanRecommendedId = designPlanRecommendedId;

}

public Integer getPlanId() {

return planId;

}

public void setPlanId(Integer planId) {

this.planId = planId;

}

public String getLogoPath() {

return logoPath;

}

public void setLogoPath(String logoPath) {

this.logoPath = logoPath;

}

public Integer getUserId() {

return userId;

}

public void setUserId(Integer userId) {

this.userId = userId;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public Integer getUserType() {

return userType;

}

public void setUserType(Integer userType) {

this.userType = userType;

}

public String getCompanyName() {

return companyName;

}

public void setCompanyName(String companyName) {

this.companyName = companyName;

}

public String getCompanyLogoPath() {

return companyLogoPath;

}

public void setCompanyLogoPath(String companyLogoPath) {

this.companyLogoPath = companyLogoPath;

}

public String getHouseTypeName() {

return houseTypeName;

}

public void setHouseTypeName(String houseTypeName) {

this.houseTypeName = houseTypeName;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getDesc() {

return desc;

}

public void setDesc(String desc) {

this.desc = desc;

}

public String getThumbUrl() {

return thumbUrl;

}

public void setThumbUrl(String thumbUrl) {

this.thumbUrl = thumbUrl;

}

public String getWindowsPercent() {

return windowsPercent;

}

public void setWindowsPercent(String windowsPercent) {

this.windowsPercent = windowsPercent;

}

}

package com.nork.render.controller;

import com.nork.cache.service.RedisService;

import com.nork.common.model.ResponseEnvelope;

import com.nork.design.model.DesignPlanRecommended;

import com.nork.design.model.DesignPlanSummaryInfo;

import com.nork.design.service.DesignPlanRecommendedServiceV2;

import com.nork.design.service.DesignPlanSummaryInfoService;

import com.nork.pano.model.scene.PanoramaVo;

import com.nork.product.model.BaseCompany;

import com.nork.product.model.CompanyShop;

import com.nork.product.service.BaseCompanyService;

import com.nork.product.service.CompanyShopService;

import com.nork.render.model.HeadMessage;

import com.nork.render.model.ResRenderData;

import com.nork.system.service.ResRenderPicService;

import org.apache.commons.lang3.StringUtils;

import org.slf4j.Logger;

import org.slf4j.LoggerFactory;

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 javax.servlet.http.HttpServletRequest;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.StringJoiner;

@Controller

@RequestMapping("/v1/miniprogram/render/renderpic")

public class ResRenderPicController {

private static final Logger logger = LoggerFactory.getLogger(ResRenderPicController.class.getName());

private final static String CLASS\_LOG\_PREFIX = "[方案资源服务]";

private final static String DESIGN\_PLAN\_VR\_VIEW\_NUM\_MAP = "CustomerDesignPlanVrViewNumMap";

private static final String DESIGN\_PLAN\_VR\_VIEW\_NUM\_SYNCHRONIZE\_MAP = "CustomerDesignPlanVrViewNumSynchronizeMap";

private static final Integer PLAN\_MODULE\_SOURCE = 1;//来自推荐方案模块的方案

private static final Integer PLAN\_SHOP\_SOURCE = 2;//来自店铺方案列表的方案

private static final Integer PLAN\_USER\_SOURCE = 3;//用户自身的方案（效果图方案）

private static final Integer SANDU\_COMPANY\_ID = 2501;//三度公司id，为了少查一次sql，就先写死了id为常量

@Autowired

private ResRenderPicService resRenderPicService;

@Autowired

private RedisService redisService;

@Autowired

private DesignPlanSummaryInfoService designPlanSummaryInfoService;

@Autowired

private DesignPlanRecommendedServiceV2 designPlanRecommendedServiceV2;

@Autowired

private BaseCompanyService baseCompanyService;

@Autowired

private CompanyShopService companyShopService;

/\*\*

\* 获取设计方案的照片,单点720,多点720,视频资源

\*

\* @param request

\* @return

\*/

@RequestMapping(value = {"/getrenderres"}, method = RequestMethod.GET)

@ResponseBody

public Object getRenderPic(@ModelAttribute PanoramaVo panoramaVo, HttpServletRequest request) {

Integer designPlanRecommendedId = panoramaVo.getDesignPlanRecommendedId();

Integer designPlanRenderSceneId = panoramaVo.getDesignPlanRenderSceneId();

long vrViewNum = 0;

Integer recommendedType = 1;

HeadMessage headMessage = new HeadMessage();

Map<String, List<ResRenderData>> map = null;

if (designPlanRecommendedId != null && designPlanRecommendedId.intValue() > 0) {

synchronized (ResRenderPicController.class){

//从缓存中获取方案720浏览数,获取不到从数据库查找,再存入缓存

String designPlanLikeNum = redisService.getMap(DESIGN\_PLAN\_VR\_VIEW\_NUM\_MAP, designPlanRecommendedId + "");

if (StringUtils.isEmpty(designPlanLikeNum)) {

DesignPlanSummaryInfo summaryInfo = designPlanSummaryInfoService.get(designPlanRecommendedId);

if (null != summaryInfo) {

if (summaryInfo.getVrViewNum() != null){

vrViewNum = (long) summaryInfo.getVrViewNum();

}

}

redisService.addMap(DESIGN\_PLAN\_VR\_VIEW\_NUM\_MAP, designPlanRecommendedId + "", vrViewNum + "");

}

vrViewNum = redisService.mapIncrby(DESIGN\_PLAN\_VR\_VIEW\_NUM\_MAP, designPlanRecommendedId + "", 1L);

//把720浏览数存到同步的缓存中

redisService.addMap(DESIGN\_PLAN\_VR\_VIEW\_NUM\_SYNCHRONIZE\_MAP, designPlanRecommendedId + "", vrViewNum + "");

}

map = resRenderPicService.getAllResRenderPic(designPlanRecommendedId, ResRenderPicService.renderResEnum.designPlanRecommended);

// 推荐方案返回是样板方案还是一件方案

DesignPlanRecommended designPlanRecommended = designPlanRecommendedServiceV2.get(designPlanRecommendedId);

if (designPlanRecommended == null) {

return new ResponseEnvelope(false,"查询到推荐方案为空，请检查参数 designPlanRecommendedId");

}

recommendedType = designPlanRecommended.getRecommendedType();

// 获取方案来源，处理头像相关信息

Integer planSource = panoramaVo.getPlanSource();

if (planSource != null && planSource != 0) {

if (PLAN\_MODULE\_SOURCE.equals(planSource)) {

// 如果是方案模块的方案，则显示厂商的头像信息，三度的方案则不显示头像信息

Integer companyId = designPlanRecommended.getCompanyId();

if (!SANDU\_COMPANY\_ID.equals(companyId)) {

// 如果不为三度的方案，则显示对应方案厂商的信息，点击跳转到品牌馆，三度的则不返回头信息

BaseCompany company = baseCompanyService.getHeadMessageOfCompany(companyId);

if (company != null) {

headMessage.setCompanyId(company.getId());

headMessage.setHeadName(company.getCompanyName());

headMessage.setHeadPath(company.getCompanyLogoPath());

// headMessage.setHeadPhoneNumber(company.getCompanyPhone());

CompanyShop companyShop = companyShopService.getCompanyShopByCompanyId(company.getId());

if (null != companyShop) {

headMessage.setShopId(companyShop.getId());

}

}

}

} else if (PLAN\_SHOP\_SOURCE.equals(planSource)) {

// 如果是店铺的推荐方案，则显示对应店铺的头像信息，并需要点击跳转对应店铺

Integer shopId = panoramaVo.getShopId();

if (shopId != null || shopId != 0) {

CompanyShop shop = baseCompanyService.getHeadMessageOfShop(shopId);

if (shop != null) {

headMessage.setShopId(shop.getId());

headMessage.setHeadName(shop.getShopName());

headMessage.setHeadPath(shop.getShopLogoPath());

}

}

}

//如果为用户自己的方案则不会走到这里，不进行处理

}

} else if (designPlanRenderSceneId != null && designPlanRenderSceneId > 0) {

map = resRenderPicService.getAllResRenderPic(designPlanRenderSceneId, ResRenderPicService.renderResEnum.designPlanRenderScene);

} else {

return new ResponseEnvelope(false, "参数方案ID为空！");

}

Map<String, Object> objectMap = new HashMap<>(map.size() + 1);

objectMap.putAll(map);

objectMap.put("pv", vrViewNum);

objectMap.put("recommendedType",recommendedType);

objectMap.put("headMessage",headMessage);

return new ResponseEnvelope(true, objectMap);

}

}

package com.nork.system.service.impl;

import java.io.File;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.Date;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import com.google.gson.reflect.TypeToken;

import com.nork.common.model.LoginUser;

import com.nork.common.model.ResponseEnvelope;

import com.nork.common.pano.PanoramaUtil;

import com.nork.common.util.FileUploadUtils;

import com.nork.common.util.JsonUtil;

import com.nork.common.util.Utils;

import com.nork.design.dao.DesignPlanMapper;

import com.nork.design.dao.DesignRenderRoamMapper;

import com.nork.design.model.\*;

import com.nork.design.service.DesignPlanRecommendedService;

import com.nork.design.service.DesignPlanRenderSceneService;

import com.nork.home.service.SpaceCommonService;

import com.nork.pano.model.output.DesignPlanStoreReleaseDetailsVo;

import com.nork.pano.model.roam.Coordinate;

import com.nork.pano.model.roam.Roam;

import com.nork.pano.model.scene.PanoramaVo;

import com.nork.pano.model.scene.SingleSceneVo;

import com.nork.product.model.BaseCompany;

import com.nork.product.service.BaseCompanyService;

import com.nork.render.model.RenderTypeCode;

import com.nork.render.model.ResRenderData;

import com.nork.system.model.\*;

import com.nork.system.model.small.ResScreenPicData;

import com.nork.system.service.\*;

import net.sf.json.JSONArray;

import org.apache.commons.lang.StringUtils;

import org.apache.log4j.Logger;

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

import org.springframework.stereotype.Service;

import com.nork.common.util.Constants;

import com.nork.design.service.DesignPlanRecommendedServiceV2;

import com.nork.home.dao.SpaceCommonMapper;

import com.nork.home.model.SpaceCommon;

import com.nork.system.dao.ResRenderPicMapper;

import com.nork.system.dao.SysDictionaryMapper;

import com.nork.system.model.search.ResRenderPicSearch;

/\*\*

\* @Title: ResRenderPicServiceImpl.java

\* @Package com.nork.system.service.impl

\* @Description:渲染图片资源库-渲染图片资源库ServiceImpl

\* @createAuthor pandajun

\* @CreateDate 2017-03-22 14:39:08

\* @version V1.0

\*/

@SuppressWarnings("all")

@Service("resRenderPicService")

public class ResRenderPicServiceImpl implements ResRenderPicService {

private static Logger logger = Logger.getLogger(ResRenderPicServiceImpl.class);

private static final String rederResPath = Utils.getPropertyName("app","app.resources.url","https://show.sanduspace.com");

private ResRenderPicMapper resRenderPicMapper;

@Autowired

private DesignPlanMapper designPlanMapper;

@Autowired

private SpaceCommonService spaceCommonService;

@Autowired

private SysDictionaryService sysDictionaryService;

@Autowired

public void setResRenderPicMapper(

ResRenderPicMapper resRenderPicMapper) {

this.resRenderPicMapper = resRenderPicMapper;

}

@Autowired

private DesignPlanRecommendedServiceV2 designPlanRecommendedServiceV2;

@Autowired

private ResRenderVideoService resRenderVideoService;

@Autowired

private DesignRenderRoamMapper designRenderRoamMapper;

@Autowired

private ResDesignService resDesignService;

@Autowired

private DesignPlanRecommendedService designPlanRecommendedService;

@Autowired

private DesignPlanRenderSceneService designPlanRenderSceneService;

@Autowired

private SysUserService sysUserService;

@Autowired

private BaseCompanyService baseCompanyService;

@Autowired

private ResPicService resPicService;

/\*\*

\* 新增数据

\*

\* @param resRenderPic

\* @return int

\*/

@Override

public int add(ResRenderPic resRenderPic) {

if(null == resRenderPic || null == resRenderPic.getBusinessId() || resRenderPic.getBusinessId().intValue()<=0)

return 0;

DesignPlan designPlan=designPlanMapper.selectByPrimaryKey(resRenderPic.getBusinessId());

if(designPlan == null){

logger.info("do not find object for id="+resRenderPic.getBusinessId());

return 0;

}

resRenderPic.setDesignSceneId(designPlan.getDesignSceneId());

resRenderPic.setDesignPlanName(designPlan.getPlanName());

SpaceCommon spaceCommon = spaceCommonService.get(designPlan.getSpaceCommonId());

logger.info("spaceCommon="+(spaceCommon==null));

if(spaceCommon!=null){

resRenderPic.setArea(spaceCommon.getSpaceAreas());//空间面积大小

SysDictionary query = new SysDictionary();

query.setType(Constants.SYS\_DICTIONARY\_TYPE\_HOUSETYPE);

query.setValue(spaceCommon.getSpaceFunctionId());

SysDictionary sysDictionary = sysDictionaryService.getNameByType(query);

logger.info("sysDictionary="+(sysDictionary!=null ));

if(sysDictionary!=null )

resRenderPic.setSpaceType(sysDictionary.getName());//设置房屋空间类型

}

logger.info(designPlan.getDesignSceneId()+"\_getSpaceType="+(resRenderPic.getSpaceType()));

resRenderPic.setCreateUserId(designPlan.getUserId());

resRenderPicMapper.insertSelective(resRenderPic);

logger.info(designPlan.getDesignSceneId()+"\_resRenderPic.getId()="+resRenderPic.getId());

return resRenderPic.getId();

}

/\*\*

\* 新增数据

\*

\* @param resRenderPic

\* @return int

\*/

@Override

public int addNew(ResRenderPic resRenderPic) {

if(null == resRenderPic || null == resRenderPic.getPlanRecommendedId() || resRenderPic.getPlanRecommendedId().intValue()<=0)

return 0;

DesignPlanRecommended designPlanRecommended = designPlanRecommendedServiceV2.get(resRenderPic.getPlanRecommendedId());

if(designPlanRecommended == null){

logger.info("do not find object for id="+resRenderPic.getPlanRecommendedId());

return 0;

}

// DesignPlan designPlan = designPlanMapper.selectByPrimaryKey(resRenderPic.getBusinessId());

// if(designPlan!= null){

// resRenderPic.setDesignSceneId(designPlan.getDesignSceneId());

// }

resRenderPic.setDesignPlanName(designPlanRecommended.getPlanName());

SpaceCommon spaceCommon = spaceCommonService.get(designPlanRecommended.getSpaceCommonId());

logger.info("spaceCommon="+(spaceCommon==null));

if(spaceCommon!=null){

resRenderPic.setArea(spaceCommon.getSpaceAreas());//空间面积大小

SysDictionary query = new SysDictionary();

query.setType(Constants.SYS\_DICTIONARY\_TYPE\_HOUSETYPE);

query.setValue(spaceCommon.getSpaceFunctionId());

SysDictionary sysDictionary = sysDictionaryService.getNameByType(query);

logger.info("sysDictionary="+(sysDictionary!=null ));

if(sysDictionary!=null )

resRenderPic.setSpaceType(sysDictionary.getName());//设置房屋空间类型

}

logger.info("getSpaceType="+(resRenderPic.getSpaceType()));

resRenderPic.setCreateUserId(designPlanRecommended.getUserId());

resRenderPicMapper.insertSelective(resRenderPic);

logger.info("resRenderPic.getId()="+resRenderPic.getId());

return resRenderPic.getId();

}

/\*\*

\* 更新数据

\*

\* @param resRenderPic

\* @return int

\*/

@Override

public int update(ResRenderPic resRenderPic) {

return resRenderPicMapper

.updateByPrimaryKeySelective(resRenderPic);

}

/\*\*

\* 删除数据

\*

\* @param id

\* @return int

\*/

@Override

public int delete(Integer id) {

return resRenderPicMapper.deleteByPrimaryKey(id);

}

/\*\*

\* 获取数据详情

\*

\* @param id

\* @return ResRenderPic

\*/

@Override

public ResRenderPic get(Integer id) {

return resRenderPicMapper.selectByPrimaryKey(id);

}

/\*\*

\* 所有数据

\*

\* @param resRenderPic

\* @return List<ResRenderPic>

\*/

@Override

public List<ResRenderPic> getList(ResRenderPic resRenderPic) {

return resRenderPicMapper.selectList(resRenderPic);

}

/\*\*

\* 获取数据数量

\*

\* @param resRenderPic

\* @return int

\*/

@Override

public int getCount(ResRenderPicSearch resRenderPicSearch){

return resRenderPicMapper.selectCount(resRenderPicSearch);

}

/\*\*

\* 分页获取数据

\*

\* @param resRenderPic

\* @return List<ResRenderPic>

\*/

@Override

public List<ResRenderPic> getPaginatedList(

ResRenderPicSearch resRenderPicSearch) {

return resRenderPicMapper.selectPaginatedList(resRenderPicSearch);

}

/\*\*

\* 其他

\*

\*/

/\*\*

\* 根据推荐方案Id和渲染类型获得最新渲染原图

\*/

@Override

public ResRenderPic getResRenderPicByPlanRecommended(Integer planRecommendedId,Integer renderingType){

if(planRecommendedId!=null){

List<ResRenderPic> renderPics=null;

renderPics=resRenderPicMapper.getResRenderPicByPlanRecommended(planRecommendedId,renderingType);

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

ResRenderPic renderPic=renderPics.get(0);

return renderPic;

}else{

return new ResRenderPic();

}

}

return new ResRenderPic();

}

/\*\*

\* 为了渲染列表查询所有的缩略图和封面图

\*/

@Override

public List<ResRenderPic> selectListByFileKeys(ResRenderPicQO resRenderPicQO) {

return resRenderPicMapper.selectListByFileKeys(resRenderPicQO);

}

@Override

public int getLatestSmallPic(ResRenderPic renderPic) {

// if (renderPic == null || StringUtils.isEmpty(renderPic.getFileKey()) || renderPic.getBusinessId() == null || renderPic.getBusinessId() == null){

// return 0;

// }

if (renderPic == null || renderPic.getFileKeyList() == null || renderPic.getFileKeyList().size() <= 0 || renderPic.getBusinessId() == null || renderPic.getBusinessId() == null){

return 0;

}

return resRenderPicMapper.getLatestSmallPicByBusi(renderPic);

}

@Override

public List<ResRenderPic> getRenderResourceListByopInfo(Integer sourcePlanId, Integer templateId,

Integer renderType) {

return resRenderPicMapper.getRenderResourceListByopInfo(sourcePlanId, templateId, renderType);

}

/\*\*

\* 通过pid获取缩略图

\* @param pid

\* @return

\*/

@Override

public ResRenderPic selectOneByPid(Integer pid){

// TODO Auto-generated method stub

return resRenderPicMapper.selectOneByPid(pid);

}

@Override

public void deletePicBySceneId(Integer sceneId) {

if(sceneId == null || sceneId < 1){

logger.error("------function:deletePicBySceneId(Integer sceneId)->sceneId == null || sceneId < 1");

return;

}

resRenderPicMapper.deletePicBySceneId(sceneId);

}

@Override

public void updateOrigPicBySmallPicId(ResRenderPic render) {

resRenderPicMapper.updateOrigPicBySmallPicId(render);

}

@Override

public int getTaskIdBySmallId(int smallId) {

return resRenderPicMapper.getTaskIdBySmallId(smallId);

}

@Override

public void updateSceneIdByTaskId(int sceneId, int taskId) {

resRenderPicMapper.updateSceneIdByTaskId(sceneId, taskId);

}

@Override

public String getQRCodeInfo(ResRenderPic resPic, Date dateDue) {

String urlSuffix = "";

// 参数验证/处理 ->start

if(dateDue != null) {

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

urlSuffix = "dateDueStr=" + simpleDateFormat.format(dateDue);

}

// 参数验证/处理 ->end

String picPath="";

String code = "";

if(resPic.getRenderingType() != null){

if( RenderTypeCode.COMMON\_720\_LEVEL == resPic.getRenderingType() || RenderTypeCode.HD\_720\_LEVEL == resPic.getRenderingType() ){// 普通单图720

ResRenderPic singleCode = resRenderPicMapper.get720SingleCode(resPic);//得到720单点分享需要的code

if(singleCode != null && StringUtils.isNotEmpty(singleCode.getSysCode())){

code = singleCode.getSysCode();

}else{

logger.error("ResRenderPic is null ,res\_render\_pic\_id="+resPic.getId());

return null;

}

picPath = "pages/vr720/vr720Single.htm?code="+code;

}else if( RenderTypeCode.ROAM\_720\_LEVEL == resPic.getRenderingType() ){// 720漫游

// 获取720漫游组信息，传过来的pid问截图ID。渲染图列表接口返回了缩略图列表和对应的截图ID

DesignRenderRoam romanCode = resRenderPicMapper.get720RomanCode(resPic);

if(romanCode != null && StringUtils.isNotEmpty(romanCode.getUuid())){

code = romanCode.getUuid();

picPath = "pages/vr720/vr720Roam.htm?code="+code;

}else{

logger.error("DesignRenderRoam is null ,res\_render\_pic\_id="+resPic.getId());

return null;

}

} else if(RenderTypeCode.COMMON\_VIDEO == resPic.getRenderingType() || RenderTypeCode.HD\_VIDEO == resPic.getRenderingType()){//普通视频

ResRenderVideo resRenderVideo = new ResRenderVideo();

resRenderVideo.setIsDeleted(0);

resRenderVideo.setSysTaskPicId(resPic.getId());

List<ResRenderVideo>videoList = resRenderVideoService.getList(resRenderVideo);

if(videoList == null || videoList.size()<=0 ){

return null;

}else if(videoList != null && videoList.size() == 1 ){

picPath = videoList.get(0).getVideoPath();

}else{

return null;

}

}else{

picPath=resPic.getPicPath();

}

}else{

/\*取普通渲染图路径\*/

picPath=resPic.getPicPath();

}

if(StringUtils.isNotEmpty(picPath) && StringUtils.isNotBlank(urlSuffix)) {

if(picPath.indexOf("?") != -1) {

picPath = picPath + "&" + urlSuffix;

}else {

picPath = picPath + "?" + urlSuffix;

}

}

return picPath;

}

@Override

public String getQRpicPath(ResRenderPic resPic, Date dateDue){

// 参数验证 ->start

// 参数验证 ->end

int renderingType = resPic.getRenderingType().intValue();

String picPath = getQRCodeInfo(resPic, dateDue);

if(resPic.getRenderingType() != null){

if( RenderTypeCode.COMMON\_720\_LEVEL == renderingType || RenderTypeCode.HD\_720\_LEVEL == renderingType ){// 普通单图720

picPath = Utils.getPropertyName("app","app.render.server.url","")

+ Utils.getPropertyName("app","app.render.server.siteName","") + picPath;

// picPath = Utils.getPropertyName("app","app.resources.url","") + picPath;

}else if( RenderTypeCode.ROAM\_720\_LEVEL == renderingType ){// 720漫游

picPath = Utils.getPropertyName("app","app.render.server.url","")

+ Utils.getPropertyName("app","app.render.server.siteName","") + picPath;

// picPath = Utils.getPropertyName("app","app.resources.url","") + picPath;

} else if(RenderTypeCode.COMMON\_VIDEO == renderingType){//普通视频

picPath = Utils.getPropertyName("app","app.resources.url","") + picPath;

}else if(RenderTypeCode.HD\_VIDEO == renderingType){//高清视频

picPath = Utils.getPropertyName("app","app.resources.url","") + picPath;

}else{

picPath=Utils.getValue("app.resources.url", "")+picPath;

}

}else{

/\*取普通渲染图路径\*/

picPath=Utils.getValue("app.resources.url", "")+picPath;

}

return picPath;

}

/\*\*

\* 通过推荐id 渲染图

\* @param planRecommendedId

\*/

@Override

public void deletedByPlanRecommendedId(int planRecommendedId) {

resRenderPicMapper.deletedByPlanRecommendedId(planRecommendedId);

}

@Override

public List<ThumbData> getRenderPicIdsBySceneId(long sceneId) {

if(sceneId < 1){

logger.error("params error, 'sceneId' <1");

return null;

}

List<ThumbData> screenIds = resRenderPicMapper.getRenderPicIdsBySceneId(sceneId);

return screenIds;

}

@Override

public int countRoamByFileKey(ResRenderPicSearch resRenderPicSearch) {

return resRenderPicMapper.countRoamByFileKey(resRenderPicSearch);

}

@Override

public int insertSelective(ResRenderPic resRenderPic) {

resRenderPicMapper.insertSelective(resRenderPic);

logger.error("resRenderPic.getId()="+resRenderPic.getId());

return resRenderPic.getId();

}

/\*\*

\* 从自动渲染里面取720全景效果图

\* @param resRenderPic

\* @return

\*/

@Override

public List<ResRenderPic> selectListOfMobile(ResRenderPic resRenderPic){

return resRenderPicMapper.selectListOfMobile(resRenderPic);

}

/\*\*

\* 从自动渲染里面通过pid取缩略图

\* @param pid

\* @return

\*/

@Override

public ResRenderPic selectOneByPidOfMobile(Integer pid){

return resRenderPicMapper.selectOneByPidOfMobile(pid);

}

@Override

public DesignRenderRoam get720RomanCode(ResRenderPic resRenderPic) {

// TODO Auto-generated method stub

return resRenderPicMapper.get720RomanCode(resRenderPic);

}

@Override

public ResRenderPic get720SingleCode(ResRenderPic resRenderPic) {

// TODO Auto-generated method stub

return resRenderPicMapper.get720SingleCode(resRenderPic);//得到720单点分享需要的code

}

/\*\*

\* 根据 联盟方案素材发布ID 查找该方案内的720单点渲染图列表

\*/

@Override

public List<ResRenderPic> getListByUnionDesignPlanStoreReleaseId(Integer releaseId) {

if(releaseId == null ) {

return null;

}

return resRenderPicMapper.getListByUnionDesignPlanStoreReleaseId(releaseId);

}

/\*\*

\* 根据图片类型和方案副本获取图片

\* @param picType

\* @param taskId

\* @return

\*/

@Override

public ResRenderPic getBytaskIdAndPicType(String picType, Integer taskId) {

if (picType == null || taskId == null) {

return null;

}

return resRenderPicMapper.getBytaskIdAndPicType(picType,taskId);

}

@Override

public Map<String, List<ResRenderData>> getAllResRenderPic(Integer id, renderResEnum type) {

Map<String, List<ResRenderData>> resRenderDataMap = new HashMap<>();

//获取照片级资源

//List<ResRenderData> photoRes = this.getPhotoRes(id, type);

//resRenderDataMap.put("photo", photoRes);

//获取视频资源

//List<ResRenderData> videoRes = this.getVideoRes(id, type);

//resRenderDataMap.put("video", videoRes);

//获取单点720资源

List<ResRenderData> single720Res = this.getSingle720Res(id, type);

resRenderDataMap.put("single720", single720Res);

//获取多点720资源

List<ResRenderData> roam720Res = this.getRoam720Res(id, type);

resRenderDataMap.put("roam720", roam720Res);

return resRenderDataMap;

}

//获取照片级资源

@SuppressWarnings("all")

private List<ResRenderData> getPhotoRes(Integer id, renderResEnum type) {

ResRenderPic resRenderPic = new ResRenderPic();

List<String> fileKeys = new ArrayList<>();

List<ResRenderData> resRenderDataList = new ArrayList<>();

if (type == renderResEnum.designPlanRecommended) {

resRenderPic.setPlanRecommendedId(id);

fileKeys.add("design.designPlanRecommended.render.small.pic");

} else if (type == renderResEnum.designPlanRenderScene) {

resRenderPic.setDesignSceneId(id);

fileKeys.add("design.designPlan.render.small.pic");

} else {

return null;

}

resRenderPic.setIsDeleted(0);

resRenderPic.setOrder("gmt\_create desc");

resRenderPic.setRenderingType(1);

resRenderPic.setFileKeyList(fileKeys);

List<ResRenderPic> picList = this.getList(resRenderPic);

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

for (ResRenderPic res : picList) {

ResRenderPic renderPic = this.get(res.getPid());

ResRenderData resRenderData = new ResRenderData();

resRenderData.setRenderingType(res.getRenderingType());

resRenderData.setPicPath(this.getRenderPicInfo(renderPic));

resRenderDataList.add(resRenderData);

}

}

return resRenderDataList;

}

//获取视频资源

private List<ResRenderData> getVideoRes(Integer id, renderResEnum type) {

ResRenderPic resRenderPic = new ResRenderPic();

List<String> fileKeys = new ArrayList<>();

List<ResRenderData> resRenderDataList = new ArrayList<>();

if (type == renderResEnum.designPlanRecommended) {

resRenderPic.setPlanRecommendedId(id);

fileKeys.add("design.designPlanRecommended.render.video.cover");

fileKeys.add("design.designPlanRecommended.render.pic");

} else if (type == renderResEnum.designPlanRenderScene) {

resRenderPic.setDesignSceneId(id);

fileKeys.add("design.designPlan.render.video.cover");

} else {

return null;

}

resRenderPic.setIsDeleted(0);

resRenderPic.setOrder("gmt\_create desc");

resRenderPic.setRenderingType(6);

resRenderPic.setFileKeyList(fileKeys);

List<ResRenderPic> picList = this.getList(resRenderPic);

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

for (ResRenderPic res : picList) {

ResRenderPic renderPic = this.get(res.getId());

ResRenderData resRenderData = new ResRenderData();

resRenderData.setRenderingType(renderPic.getRenderingType());

resRenderData.setPicPath(this.getRenderPicInfo(renderPic));

resRenderDataList.add(resRenderData);

}

}

return resRenderDataList;

}

//获取单点720资源

@SuppressWarnings("all")

private List<ResRenderData> getSingle720Res(Integer id, renderResEnum type) {

ResRenderPic resRenderPic = new ResRenderPic();

List<String> fileKeys = new ArrayList<>();

List<ResRenderData> resRenderDataList = new ArrayList<>();

if (type == renderResEnum.designPlanRecommended) {

resRenderPic.setPlanRecommendedId(id);

fileKeys.add("design.designPlanRecommended.render.small.pic");

} else if (type == renderResEnum.designPlanRenderScene) {

resRenderPic.setDesignSceneId(id);

fileKeys.add("design.designPlan.render.small.pic");

} else {

return null;

}

resRenderPic.setIsDeleted(0);

resRenderPic.setOrder("gmt\_create desc");

resRenderPic.setRenderingType(4);

resRenderPic.setFileKeyList(fileKeys);

List<ResRenderPic> picList = this.getList(resRenderPic);

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

for (ResRenderPic res : picList) {

ResRenderPic renderPic = this.get(res.getPid());

ResRenderData resRenderData = new ResRenderData();

resRenderData.setRenderingType(res.getRenderingType());

resRenderData.setPicPath(this.getRenderPicInfo(renderPic));

SingleSceneVo singleSceneVo = new SingleSceneVo();

//获取单点720缩略图和原图

singleSceneVo.setThumbnailPicPath(res.getPicPath());

singleSceneVo.setPicPath(renderPic.getPicPath());

// 如果图片资源为目录，则表示为切片资源

File file = new File(Utils.getAbsolutePath(renderPic.getPicPath(), null));

if (file.exists() && file.isDirectory()) {

logger.error("DesignPlanStoreReleaseDetailsVo.IsShear.YES ： " + 1);

singleSceneVo.setIsShear(DesignPlanStoreReleaseDetailsVo.IsShear.YES);

}

// 获取页面需要显示的业务信息（用户昵称、企业LOGO、企业名称

PanoramaVo panoramaVo = new PanoramaVo();

PanoramaVo newPanoramaVo = getUserInfoDetail(id, panoramaVo, type);

singleSceneVo.setPanoramaVo(newPanoramaVo);

resRenderData.setSingleSceneVo(singleSceneVo);

resRenderDataList.add(resRenderData);

}

}

return resRenderDataList;

}

//获取多点720资源

@SuppressWarnings("all")

private List<ResRenderData> getRoam720Res(Integer id, renderResEnum type) {

ResRenderPic resRenderPic = new ResRenderPic();

List<String> fileKeys = new ArrayList<>();

List<ResRenderData> resRenderDataList = new ArrayList<>();

if (type == renderResEnum.designPlanRecommended) {

resRenderPic.setPlanRecommendedId(id);

fileKeys.add("design.designPlanRecommended.render.small.pic");

} else if (type == renderResEnum.designPlanRenderScene) {

resRenderPic.setDesignSceneId(id);

fileKeys.add("design.designPlan.render.small.pic");

} else {

return null;

}

resRenderPic.setIsDeleted(0);

resRenderPic.setOrder("gmt\_create desc");

resRenderPic.setRenderingType(8);

resRenderPic.setFileKeyList(fileKeys);

List<ResRenderPic> picList = this.getList(resRenderPic);

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

for (ResRenderPic res : picList) {

ResRenderPic renderPic = null;

if (type == renderResEnum.designPlanRecommended) {

renderPic = this.get(res.getPid());

} else if (type == renderResEnum.designPlanRenderScene) {

renderPic = this.get(res.getSysTaskPicId());

} else {

return null;

}

ResRenderData resRenderData = new ResRenderData();

resRenderData.setRenderingType(res.getRenderingType());

resRenderData.setPicPath(this.getRenderPicInfo(renderPic));

SingleSceneVo singleSceneVo = new SingleSceneVo();

//获取多点720的渲染截图

singleSceneVo.setPicPath(renderPic.getPicPath());

singleSceneVo.setThumbnailPicPath(res.getPicPath());

//获取多点720的热点位置

List<Roam> roams = getRoams(resRenderData);

if (null != roams && 0 <roams.size()) {

singleSceneVo.setRoamList(roams);

} else {

//推荐方案720多点容错处理xiaoxc\_20180815

//场景：多点没有配置文件无法打开720多点图片，容错显示单个图片

if (type == renderResEnum.designPlanRecommended) {

String picPath = resRenderPicMapper.getRenderRoamPicByRecommendedPlanId(id);

if (!StringUtils.isEmpty(picPath)) {

singleSceneVo.setPicPath(picPath);

// 如果图片资源为目录，则表示为切片资源

File file = new File(Utils.getAbsolutePath(picPath, null));

if (file.exists() && file.isDirectory()) {

singleSceneVo.setIsShear(DesignPlanStoreReleaseDetailsVo.IsShear.YES);

}

}

}

}

// 获取页面需要显示的业务信息（用户昵称、企业LOGO、企业名称

PanoramaVo panoramaVo = new PanoramaVo();

PanoramaVo newPanoramaVo = getUserInfoDetail(id, panoramaVo, type);

singleSceneVo.setPanoramaVo(newPanoramaVo);

resRenderData.setSingleSceneVo(singleSceneVo);

resRenderDataList.add(resRenderData);

}

}

return resRenderDataList;

}

private String getRenderPicInfo(ResRenderPic resPic) {

String picPath;

if (resPic.getRenderingType() != null) {

if (RenderTypeCode.COMMON\_720\_LEVEL == resPic.getRenderingType() || RenderTypeCode.HD\_720\_LEVEL == resPic.getRenderingType()) {// 普通单图720

ResRenderPic singleCode = resRenderPicMapper.get720SingleCode(resPic);//得到720单点分享需要的code

if (singleCode != null && org.apache.commons.lang3.StringUtils.isNotEmpty(singleCode.getSysCode())) {

picPath = singleCode.getSysCode();

} else {

return null;

}

} else if (RenderTypeCode.ROAM\_720\_LEVEL == resPic.getRenderingType()) {// 720漫游

// 获取720漫游组信息，传过来的pid问截图ID。渲染图列表接口返回了缩略图列表和对应的截图ID

DesignRenderRoam romanCode = resRenderPicMapper.get720RomanCode(resPic);

if (romanCode != null && org.apache.commons.lang3.StringUtils.isNotEmpty(romanCode.getUuid())) {

picPath = romanCode.getUuid();

} else {

return null;

}

} else if (RenderTypeCode.COMMON\_VIDEO == resPic.getRenderingType() || RenderTypeCode.HD\_VIDEO == resPic.getRenderingType()) {//普通视频

ResRenderVideo resRenderVideo = new ResRenderVideo();

resRenderVideo.setIsDeleted(0);

resRenderVideo.setSysTaskPicId(resPic.getId());

List<ResRenderVideo> videoList = resRenderVideoService.getList(resRenderVideo);

if (videoList == null || videoList.size() <= 0) {

return null;

} else if (videoList != null && videoList.size() == 1) {

picPath = videoList.get(0).getVideoPath();

} else {

return null;

}

} else {

picPath = resPic.getPicPath();

}

} else {

/\*取普通渲染图路径\*/

picPath = resPic.getPicPath();

}

return picPath;

}

private List<Roam> getRoams(ResRenderData resRenderData) {

List<Roam> roamList = null;

DesignRenderRoam designRenderRoam = designRenderRoamMapper.selectByUUID(resRenderData.getPicPath());

if (designRenderRoam != null) {

// 获取多720图片之间的关系信息

String roamConfigContext = "";

Integer roamConfigId = designRenderRoam.getRoamConfig();

if (roamConfigId != null && roamConfigId.intValue() > 0) {

ResDesign roamConfig = resDesignService.get(roamConfigId);

if (roamConfig != null) {

String filePath = rederResPath + roamConfig.getFilePath();

roamConfigContext = FileUploadUtils.getJsonByInternet(filePath);

}

}

logger.error("roamConfigContext = " + roamConfigContext);

if (StringUtils.isNotBlank(roamConfigContext)) {

JSONArray jsonArray = JSONArray.fromObject(roamConfigContext);

roamList = JSONArray.toList(jsonArray, Roam.class);

// roamList = JsonUtil.fromJson(roamConfigContext, new TypeToken<Roam>() {}.getType());

}

}

List<Roam> roams = new ArrayList<>();

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

for (Roam roam : roamList) {

Integer resId = roam.getFieldName();

ResRenderPic r = this.get(resId);

List<Coordinate> coordinateList = new ArrayList<>();

roam.setFieldNamePath(r.getPicPath());

// 如果图片资源为目录，则表示为切片资源

File file = new File(Utils.getAbsolutePath(r.getPicPath(), null));

if( file.exists() && file.isDirectory() ){

roam.setIsShear(DesignPlanStoreReleaseDetailsVo.IsShear.YES);

}

for (Roam roam1 : roamList) {

Integer resId1 = roam1.getFieldName();

ResRenderPic r1 = this.get(resId1);

Coordinate coordinate = new Coordinate();

if (roam1.getFieldName() == roam.getFieldName()) {

continue;

}

coordinate.setAth(PanoramaUtil.getAth(roam, roam1));

coordinate.setAtv(PanoramaUtil.getAtv(roam, roam1));

coordinate.setTargetFieldName(roam1.getFieldName());

coordinate.setTargetFieldNamePath(r1.getPicPath());

coordinateList.add(coordinate);

}

roam.setCoordinateList(coordinateList);

roams.add(roam);

}

}

return roams;

}

private PanoramaVo getUserInfoDetail(Integer id, PanoramaVo panoramaVo, renderResEnum type) {

if (type == renderResEnum.designPlanRecommended) {

DesignPlanRecommended designPlanRecommended = designPlanRecommendedService.get(id);

panoramaVo = getPanoramaVoByRecommendedId(panoramaVo, designPlanRecommended);

} else if (type == renderResEnum.designPlanRenderScene) {

DesignPlanRenderScene designPlanRenderScene = designPlanRenderSceneService.get(id);

panoramaVo = getPanoramaVoBySceneId(panoramaVo, designPlanRenderScene);

} else {

return null;

}

return panoramaVo;

}

@SuppressWarnings("all")

private PanoramaVo getPanoramaVoByRecommendedId(PanoramaVo panoramaVo, DesignPlanRecommended designPlanRecommended) {

panoramaVo.setUserId(designPlanRecommended.getUserId());

Integer userId = designPlanRecommended.getUserId();

SysUser sysUser = sysUserService.get(userId);

if (sysUser != null) {

panoramaVo.setUserId(sysUser.getId());

panoramaVo.setUserType(sysUser.getUserType());

panoramaVo.setUserName(sysUser.getUserName());

//如果是经销商账号 就取公司的Pid

if (sysUser.getUserType().intValue() == 3) {

// 获取用户公司

Integer companyId = sysUser.getBusinessAdministrationId();

BaseCompany company = baseCompanyService.get(companyId);

if (company.getPid().intValue() > 0) {

BaseCompany baseCompany = baseCompanyService.get(company.getPid());

if (baseCompany != null) {

panoramaVo.setCompanyName(baseCompany.getCompanyName());

// 获取用户公司LOGO

if (baseCompany.getCompanyLogo() != null) {

ResPic resPic = resPicService.get(baseCompany.getCompanyLogo());

if (resPic != null) {

panoramaVo.setCompanyLogoPath(resPic.getPicPath());

}

}

}

}

} else {

// 获取用户公司

Integer companyId = sysUser.getBusinessAdministrationId();

BaseCompany baseCompany = baseCompanyService.get(companyId);

if (baseCompany != null) {

panoramaVo.setCompanyName(baseCompany.getCompanyName());

// 获取用户公司LOGO

if (baseCompany.getCompanyLogo() != null) {

ResPic resPic = resPicService.get(baseCompany.getCompanyLogo());

if (resPic != null) {

panoramaVo.setCompanyLogoPath(Utils.getAbsoluteUrlByRelativeUrl(resPic.getPicPath()));

}

}

}

}

}

// 获取房型类型

if (designPlanRecommended.getSpaceCommonId() != null) {

SpaceCommon spaceCommon = spaceCommonService.get(designPlanRecommended.getSpaceCommonId());

if (spaceCommon != null && spaceCommon.getSpaceFunctionId() != null) {

SysDictionary sysDictionary = sysDictionaryService.findOneByTypeAndValue("houseType", spaceCommon.getSpaceFunctionId());

panoramaVo.setHouseTypeName("我设计的" + sysDictionary.getName());

}

}

return panoramaVo;

}

/\*\*

\* 根据场景ID获取全景vo信息

\* @param panoramaVo

\* @param designPlanRenderScene

\* @return

\*/

@SuppressWarnings("all")

private PanoramaVo getPanoramaVoBySceneId(PanoramaVo panoramaVo, DesignPlanRenderScene designPlanRenderScene) {

panoramaVo.setUserId(designPlanRenderScene.getUserId());

Integer userId = designPlanRenderScene.getUserId();

SysUser sysUser = sysUserService.get(userId);

if (sysUser != null) {

panoramaVo.setUserId(sysUser.getId());

panoramaVo.setUserType(sysUser.getUserType());

panoramaVo.setUserName(sysUser.getUserName());

//如果是经销商账号 就取公司的Pid

if (sysUser.getUserType().intValue() == 3) {

// 获取用户公司

Integer companyId = sysUser.getBusinessAdministrationId();

BaseCompany company = baseCompanyService.get(companyId);

if (company.getPid().intValue() > 0) {

BaseCompany baseCompany = baseCompanyService.get(company.getPid());

if (baseCompany != null) {

panoramaVo.setCompanyName(baseCompany.getCompanyName());

// 获取用户公司LOGO

if (baseCompany.getCompanyLogo() != null) {

ResPic resPic = resPicService.get(baseCompany.getCompanyLogo());

if (resPic != null) {

panoramaVo.setCompanyLogoPath(resPic.getPicPath());

}

}

}

}

} else {

// 获取用户公司

Integer companyId = sysUser.getBusinessAdministrationId();

BaseCompany baseCompany = baseCompanyService.get(companyId);

if (baseCompany != null) {

panoramaVo.setCompanyName(baseCompany.getCompanyName());

// 获取用户公司LOGO

if (baseCompany.getCompanyLogo() != null) {

ResPic resPic = resPicService.get(baseCompany.getCompanyLogo());

if (resPic != null) {

panoramaVo.setCompanyLogoPath(Utils.getAbsoluteUrlByRelativeUrl(resPic.getPicPath()));

}

}

}

}

}

// 获取房型类型

if (designPlanRenderScene.getSpaceCommonId() != null) {

SpaceCommon spaceCommon = spaceCommonService.get(designPlanRenderScene.getSpaceCommonId());

if (spaceCommon != null && spaceCommon.getSpaceFunctionId() != null) {

SysDictionary sysDictionary = sysDictionaryService.findOneByTypeAndValue("houseType", spaceCommon.getSpaceFunctionId());

panoramaVo.setHouseTypeName("我设计的" + sysDictionary.getName());

}

}

return panoramaVo;

}

}

package com.nork.system.model;

import java.io.Serializable;

import java.util.ArrayList;

import java.util.Date;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import com.nork.common.model.Mapper;

import com.nork.mobile.model.ProductGroupReplaceTaskDetail;

import com.nork.mobile.model.ProductReplaceTaskDetail;

/\*\*

\* @Title: ResRenderPic.java

\* @Package com.nork.system.model

\* @Description:渲染图片资源库-渲染图片资源库

\* @createAuthor pandajun

\* @CreateDate 2017-03-22 14:39:08

\* @version V1.0

\*/

public class ResRenderPic extends Mapper implements Serializable{

private static final long serialVersionUID = 1L;

private Integer id;

public Integer getId() {

return id;

}

public void setId(Integer id) {

this.id = id;

}

/\*\* 设计方案 \*\*/

private Integer businessId;

/\*\* 图片编码 \*\*/

private String picCode;

/\*\* 图片名称 \*\*/

private String picName;

/\*\* 图片文件名称 \*\*/

private String picFileName;

/\*\* 图片类型 \*\*/

private String picType;

/\*\* 图片大小 \*\*/

private Integer picSize;

/\*\* 图片长 \*\*/

private Integer picWeight;

/\*\* 图片高 \*\*/

private Integer picHigh;

/\*\* 图片后缀 \*\*/

private String picSuffix;

/\*\* 图片级别 \*\*/

private Integer picLevel;

/\*\* 图片格式 \*\*/

private String picFormat;

/\*\* 图片路径 \*\*/

private String picPath;

/\*\* 图片描述 \*\*/

private String picDesc;

/\*\* 图片排序 \*\*/

private Integer picOrdering;

/\*\* key标识 \*\*/

private String fileKey;

/\*\* key标识（多个） \*\*/

private String fileKeys;

/\*\* 图片对应的缩略图id信息 \*\*/

private String smallPicInfo;

/\*\* 渲染图视角 \*\*/

private Integer viewPoint;

/\*\* 渲染图场景 \*\*/

private Integer scene;

/\*\* 渲染类型 \*\*/

private Integer renderingType;

/\*\* 图片文件排序序号 \*\*/

private Integer sequence;

/\*\* 全景图地址 \*\*/

private String panoPath;

/\*\* 渲染任务创建时间 \*\*/

private Date taskCreateTime;

/\*\* 原图id \*\*/

private Integer pid;

/\*\* 系统编码 \*\*/

private String sysCode;

/\*\* 创建者 \*\*/

private String creator;

/\*\* 创建时间 \*\*/

private Date gmtCreate;

/\*\* 修改人 \*\*/

private String modifier;

/\*\* 修改时间 \*\*/

private Date gmtModified;

/\*\* 是否删除 \*\*/

private Integer isDeleted;

/\*\* 字符备用1 \*\*/

private String att1;

/\*\* 字符备用2 \*\*/

private String att2;

/\*\* 整数备用1 \*\*/

private Integer numa1;

/\*\* 整数备用2 \*\*/

private Integer numa2;

/\*\* 备注 \*\*/

private String remark;

private String OriginalPicPath;

/\*\* 原图ID。缩略图时使用 \*\*/

private Integer baseRenderId;

/\*\*

\* 渲染任务截图

\*/

private Integer sysTaskPicId;

/\*\* 720全景图清晰度等级 \*\*/

private Integer panoLevel;

/\*\* 漫游图片关系 \*\*/

private String roam;

/\*\*

\* 设计方案副本id

\*/

private Integer designSceneId;

private Integer sourcePlanId;

private Integer templateId;

List<String>fileKeyList = new ArrayList<String>();

private String renderTaskType;

private Integer userId;

/\*\*

\* 订单产品类型(具体可以见ProductType.java)

\*/

private String productType;

/\*\*

\* 订单总金额，单位为分

\*/

private Double totalFee;

/\*\*

\* 支付类型

\*/

private String payType;

//移动端我的替换列表需要这个值作为参数

//现在全屋方案制作需要这个值，作为全屋方案主任务id，逻辑分开处理，可以共用 add by gaoj 2018.09.21

private Integer taskId;

private List<Integer> recommendIds;

private List<Integer> sceneIds;

private Integer houseId;

/\*\*用户名称\*/

private String userName;

/\*\*

\* 推荐方案id

\*/

private Integer recommendedPlanId;

/\*\*

\* 移动端一键替换的产品集合

\*/

private List<ProductReplaceTaskDetail> productReplaceList;

/\*\*

\* 移动端产品移除

\*/

private List<ProductReplaceTaskDetail> productDeleteList;

/\*\*

\* 移动端组合产品替换

\*/

private List<ProductGroupReplaceTaskDetail> productGroupReplaceList;

/\*\*

\* 移动端材质替换

\*/

private List<ProductReplaceTaskDetail> textureReplaceList;

private String orderNo;

/\*\*任务来源 （1 移动端 0 web端）\*/

private Integer taskSource;

/\*\*0装进我家 1 替换渲染\*/

private Integer taskType;

private String token;

private String platformCode;

/\*\* 出现重叠时的解决方案 \*\*/

private Integer overlapResolve;

/\*\* 渲染相机位置 \*\*/

private String cameraLocation;

/\*\*方案空间类型（1：单空间方案，2：全屋方案,3:全屋方案新逻辑）\*\*/

private Integer planHouseType;

/\*\*全屋方案主方案id\*\*/

private Integer fullHousePlanId;

/\*\*方案打组主方案id\*\*/

private Integer groupPrimaryId;

/\*\*新全屋方案id\*\*/

private Integer newFullHousePlanId;

/\*\*原效果图方案id，全屋方案替换某个效果图方案\*\*/

private Integer preRenderSceneId;

/\*\*全屋方案操作(1:创建全屋方案,2:修改全屋方案,3:修改全屋方案生成新的全屋方案,4:处理复制过来的全屋方案【有全屋方案，没有主子任务】)\*\*/

private Integer fullHousePlanAction;

/\*\*空间类型\*\*/

private Integer spaceFunctionId;

/\*全屋户型坐标信息ID\*/

private Integer houseGuidePicInfoId;

private Integer designTempletId;

/\*业务类型,1是单空间装进我家,2是全屋替换\*/

private Integer bizType;

/\*全屋户型坐标信息ID集合\*/

private List<Integer> houseGuidePicInfoIdList;

/\*\*供求信息id\*\*/

private Integer supplyDemandId;

public Integer getSupplyDemandId() {

return supplyDemandId;

}

public void setSupplyDemandId(Integer supplyDemandId) {

this.supplyDemandId = supplyDemandId;

}

public List<Integer> getHouseGuidePicInfoIdList() {

return houseGuidePicInfoIdList;

}

public void setHouseGuidePicInfoIdList(List<Integer> houseGuidePicInfoIdList) {

this.houseGuidePicInfoIdList = houseGuidePicInfoIdList;

}

public Integer getBizType() {

return bizType;

}

public void setBizType(Integer bizType) {

this.bizType = bizType;

}

public Integer getDesignTempletId() {

return designTempletId;

}

public void setDesignTempletId(Integer designTempletId) {

this.designTempletId = designTempletId;

}

public Integer getHouseGuidePicInfoId() {

return houseGuidePicInfoId;

}

public void setHouseGuidePicInfoId(Integer houseGuidePicInfoId) {

this.houseGuidePicInfoId = houseGuidePicInfoId;

}

public Integer getSpaceFunctionId() {

return spaceFunctionId;

}

public void setSpaceFunctionId(Integer spaceFunctionId) {

this.spaceFunctionId = spaceFunctionId;

}

public Integer getNewFullHousePlanId() {

return newFullHousePlanId;

}

public void setNewFullHousePlanId(Integer newFullHousePlanId) {

this.newFullHousePlanId = newFullHousePlanId;

}

public Integer getPreRenderSceneId() {

return preRenderSceneId;

}

public void setPreRenderSceneId(Integer preRenderSceneId) {

this.preRenderSceneId = preRenderSceneId;

}

public Integer getFullHousePlanAction() {

return fullHousePlanAction;

}

public void setFullHousePlanAction(Integer fullHousePlanAction) {

this.fullHousePlanAction = fullHousePlanAction;

}

public Integer getGroupPrimaryId() {

return groupPrimaryId;

}

public void setGroupPrimaryId(Integer groupPrimaryId) {

this.groupPrimaryId = groupPrimaryId;

}

public Integer getPlanHouseType() {

return planHouseType;

}

public void setPlanHouseType(Integer planHouseType) {

this.planHouseType = planHouseType;

}

public Integer getFullHousePlanId() {

return fullHousePlanId;

}

public void setFullHousePlanId(Integer fullHousePlanId) {

this.fullHousePlanId = fullHousePlanId;

}

private String payMethod;

private Integer useType;

private String companyDomainName;

public String getCompanyDomainName() {

return companyDomainName;

}

public void setCompanyDomainName(String companyDomainName) {

this.companyDomainName = companyDomainName;

}

public Integer getUseType() {

return useType;

}

public void setUseType(Integer useType) {

this.useType = useType;

}

public String getPayMethod() {

return payMethod;

}

public void setPayMethod(String payMethod) {

this.payMethod = payMethod;

}

public String getPlatformCode() {

return platformCode;

}

public void setPlatformCode(String platformCode) {

this.platformCode = platformCode;

}

public String getToken() {

return token;

}

public void setToken(String token) {

this.token = token;

}

public Integer getTaskSource() {

return taskSource;

}

public void setTaskSource(Integer taskSource) {

this.taskSource = taskSource;

}

public Integer getTaskType() {

return taskType;

}

public void setTaskType(Integer taskType) {

this.taskType = taskType;

}

public String getOrderNo() {

return orderNo;

}

public void setOrderNo(String orderNo) {

this.orderNo = orderNo;

}

public Integer getRecommendedPlanId() {

return recommendedPlanId;

}

public void setRecommendedPlanId(Integer recommendedPlanId) {

this.recommendedPlanId = recommendedPlanId;

}

public List<ProductReplaceTaskDetail> getProductReplaceList() {

return productReplaceList;

}

public void setProductReplaceList(List<ProductReplaceTaskDetail> productReplaceList) {

this.productReplaceList = productReplaceList;

}

public List<ProductReplaceTaskDetail> getProductDeleteList() {

return productDeleteList;

}

public void setProductDeleteList(List<ProductReplaceTaskDetail> productDeleteList) {

this.productDeleteList = productDeleteList;

}

public List<ProductGroupReplaceTaskDetail> getProductGroupReplaceList() {

return productGroupReplaceList;

}

public void setProductGroupReplaceList(List<ProductGroupReplaceTaskDetail> productGroupReplaceList) {

this.productGroupReplaceList = productGroupReplaceList;

}

public List<ProductReplaceTaskDetail> getTextureReplaceList() {

return textureReplaceList;

}

public void setTextureReplaceList(List<ProductReplaceTaskDetail> textureReplaceList) {

this.textureReplaceList = textureReplaceList;

}

public Integer getHouseId() {

return houseId;

}

public void setHouseId(Integer houseId) {

this.houseId = houseId;

}

public String getUserName() {

return userName;

}

public void setUserName(String userName) {

this.userName = userName;

}

public List<Integer> getRecommendIds() {

return recommendIds;

}

public void setRecommendIds(List<Integer> recommendIds) {

this.recommendIds = recommendIds;

}

public List<Integer> getSceneIds() {

return sceneIds;

}

public void setSceneIds(List<Integer> sceneIds) {

this.sceneIds = sceneIds;

}

public Integer getTaskId() {

return taskId;

}

public void setTaskId(Integer taskId) {

this.taskId = taskId;

}

public String getProductType() {

return productType;

}

public void setProductType(String productType) {

this.productType = productType;

}

public Double getTotalFee() {

return totalFee;

}

public void setTotalFee(Double totalFee) {

this.totalFee = totalFee;

}

public String getPayType() {

return payType;

}

public void setPayType(String payType) {

this.payType = payType;

}

public String getRenderTaskType() {

return renderTaskType;

}

public void setRenderTaskType(String renderTaskType) {

this.renderTaskType = renderTaskType;

}

public Integer getUserId() {

return userId;

}

public void setUserId(Integer userId) {

this.userId = userId;

}

public List<String> getFileKeyList() {

return fileKeyList;

}

public void setFileKeyList(List<String> fileKeyList) {

this.fileKeyList = fileKeyList;

}

public Integer getSourcePlanId() {

return sourcePlanId;

}

public void setSourcePlanId(Integer sourcePlanId) {

this.sourcePlanId = sourcePlanId;

}

public Integer getTemplateId() {

return templateId;

}

public void setTemplateId(Integer templateId) {

this.templateId = templateId;

}

/\*\*

\* 渲染时的实际方案名称，后面设计方案发生修改，名称不在同步

\*/

private String designPlanName;

/\*\*

\* 设计方案对应的房屋空间类型

\*/

private String spaceType;

/\*\*

\* 面积范围

\*/

private String area;

/\*\*

\* 创建者的userid

\*/

private Integer planRecommendedId;

//操作入口 0：我的设计 1：推荐方案

private Integer operationSource;

private Integer designPlanSceneId;

public Integer getDesignPlanSceneId() {

return designPlanSceneId;

}

public void setDesignPlanSceneId(Integer designPlanSceneId) {

this.designPlanSceneId = designPlanSceneId;

}

public Integer getOperationSource() {

return operationSource;

}

public void setOperationSource(Integer operationSource) {

this.operationSource = operationSource;

}

public Integer getPlanRecommendedId() {

return planRecommendedId;

}

public void setPlanRecommendedId(Integer planRecommendedId) {

this.planRecommendedId = planRecommendedId;

}

private long createUserId;

public Integer getDesignSceneId() {

return designSceneId;

}

public void setDesignSceneId(Integer designSceneId) {

this.designSceneId = designSceneId;

}

public String getOriginalPicPath() {

return OriginalPicPath;

}

public void setOriginalPicPath(String originalPicPath) {

OriginalPicPath = originalPicPath;

}

public Integer getBaseRenderId() {

return baseRenderId;

}

public void setBaseRenderId(Integer baseRenderId) {

this.baseRenderId = baseRenderId;

}

public Integer getSysTaskPicId() {

return sysTaskPicId;

}

public void setSysTaskPicId(Integer sysTaskPicId) {

this.sysTaskPicId = sysTaskPicId;

}

public Integer getBusinessId() {

return businessId;

}

public void setBusinessId(Integer businessId){

this.businessId = businessId;

}

public String getPicCode() {

return picCode;

}

public void setPicCode(String picCode){

this.picCode = picCode;

}

public String getPicName() {

return picName;

}

public void setPicName(String picName){

this.picName = picName;

}

public String getPicFileName() {

return picFileName;

}

public void setPicFileName(String picFileName){

this.picFileName = picFileName;

}

public String getPicType() {

return picType;

}

public void setPicType(String picType){

this.picType = picType;

}

public Integer getPicSize() {

return picSize;

}

public void setPicSize(Integer picSize){

this.picSize = picSize;

}

public Integer getPicWeight() {

return picWeight;

}

public void setPicWeight(Integer picWeight){

this.picWeight = picWeight;

}

public Integer getPicHigh() {

return picHigh;

}

public void setPicHigh(Integer picHigh){

this.picHigh = picHigh;

}

public String getPicSuffix() {

return picSuffix;

}

public void setPicSuffix(String picSuffix){

this.picSuffix = picSuffix;

}

public Integer getPicLevel() {

return picLevel;

}

public void setPicLevel(Integer picLevel){

this.picLevel = picLevel;

}

public String getPicFormat() {

return picFormat;

}

public void setPicFormat(String picFormat){

this.picFormat = picFormat;

}

public String getPicPath() {

return picPath;

}

public void setPicPath(String picPath){

this.picPath = picPath;

}

public String getPicDesc() {

return picDesc;

}

public void setPicDesc(String picDesc){

this.picDesc = picDesc;

}

public Integer getPicOrdering() {

return picOrdering;

}

public void setPicOrdering(Integer picOrdering){

this.picOrdering = picOrdering;

}

public String getFileKey() {

return fileKey;

}

public void setFileKey(String fileKey){

this.fileKey = fileKey;

}

public String getFileKeys() {

return fileKeys;

}

public void setFileKeys(String fileKeys){

this.fileKeys = fileKeys;

}

public String getSmallPicInfo() {

return smallPicInfo;

}

public void setSmallPicInfo(String smallPicInfo){

this.smallPicInfo = smallPicInfo;

}

public Integer getViewPoint() {

return viewPoint;

}

public void setViewPoint(Integer viewPoint){

this.viewPoint = viewPoint;

}

public Integer getScene() {

return scene;

}

public void setScene(Integer scene){

this.scene = scene;

}

public Integer getRenderingType() {

return renderingType;

}

public void setRenderingType(Integer renderingType){

this.renderingType = renderingType;

}

public Integer getSequence() {

return sequence;

}

public void setSequence(Integer sequence){

this.sequence = sequence;

}

public String getPanoPath() {

return panoPath;

}

public void setPanoPath(String panoPath){

this.panoPath = panoPath;

}

public Date getTaskCreateTime() {

return taskCreateTime;

}

public void setTaskCreateTime(Date taskCreateTime){

this.taskCreateTime = taskCreateTime;

}

public Integer getPid() {

return pid;

}

public void setPid(Integer pid){

this.pid = pid;

}

public String getSysCode() {

return sysCode;

}

public void setSysCode(String sysCode){

this.sysCode = sysCode;

}

public String getCreator() {

return creator;

}

public void setCreator(String creator){

this.creator = creator;

}

public Date getGmtCreate() {

return gmtCreate;

}

public void setGmtCreate(Date gmtCreate){

this.gmtCreate = gmtCreate;

}

public String getModifier() {

return modifier;

}

public void setModifier(String modifier){

this.modifier = modifier;

}

public Date getGmtModified() {

return gmtModified;

}

public void setGmtModified(Date gmtModified){

this.gmtModified = gmtModified;

}

public Integer getIsDeleted() {

return isDeleted;

}

public void setIsDeleted(Integer isDeleted){

this.isDeleted = isDeleted;

}

public String getAtt1() {

return att1;

}

public void setAtt1(String att1){

this.att1 = att1;

}

public String getAtt2() {

return att2;

}

public void setAtt2(String att2){

this.att2 = att2;

}

public Integer getNuma1() {

return numa1;

}

public void setNuma1(Integer numa1){

this.numa1 = numa1;

}

public Integer getNuma2() {

return numa2;

}

public void setNuma2(Integer numa2){

this.numa2 = numa2;

}

public String getRemark() {

return remark;

}

public void setRemark(String remark){

this.remark = remark;

}

public Integer getPanoLevel() {

return panoLevel;

}

public void setPanoLevel(Integer panoLevel) {

this.panoLevel = panoLevel;

}

public String getRoam() {

return roam;

}

public void setRoam(String roam) {

this.roam = roam;

}

@Override

public boolean equals(Object that) {

if (this == that) {

return true;

}

if (that == null) {

return false;

}

if (getClass() != that.getClass()) {

return false;

}

ResRenderPic other = (ResRenderPic) that;

return (this.getId() == null ? other.getId() == null : this.getId().equals(other.getId()))

&& (this.getBusinessId() == null ? other.getBusinessId() == null : this.getBusinessId().equals(other.getBusinessId()))

&& (this.getPicCode() == null ? other.getPicCode() == null : this.getPicCode().equals(other.getPicCode()))

&& (this.getPicName() == null ? other.getPicName() == null : this.getPicName().equals(other.getPicName()))

&& (this.getPicFileName() == null ? other.getPicFileName() == null : this.getPicFileName().equals(other.getPicFileName()))

&& (this.getPicType() == null ? other.getPicType() == null : this.getPicType().equals(other.getPicType()))

&& (this.getPicSize() == null ? other.getPicSize() == null : this.getPicSize().equals(other.getPicSize()))

&& (this.getPicWeight() == null ? other.getPicWeight() == null : this.getPicWeight().equals(other.getPicWeight()))

&& (this.getPicHigh() == null ? other.getPicHigh() == null : this.getPicHigh().equals(other.getPicHigh()))

&& (this.getPicSuffix() == null ? other.getPicSuffix() == null : this.getPicSuffix().equals(other.getPicSuffix()))

&& (this.getPicLevel() == null ? other.getPicLevel() == null : this.getPicLevel().equals(other.getPicLevel()))

&& (this.getPicFormat() == null ? other.getPicFormat() == null : this.getPicFormat().equals(other.getPicFormat()))

&& (this.getPicPath() == null ? other.getPicPath() == null : this.getPicPath().equals(other.getPicPath()))

&& (this.getPicDesc() == null ? other.getPicDesc() == null : this.getPicDesc().equals(other.getPicDesc()))

&& (this.getPicOrdering() == null ? other.getPicOrdering() == null : this.getPicOrdering().equals(other.getPicOrdering()))

&& (this.getFileKey() == null ? other.getFileKey() == null : this.getFileKey().equals(other.getFileKey()))

&& (this.getFileKeys() == null ? other.getFileKeys() == null : this.getFileKeys().equals(other.getFileKeys()))

&& (this.getSmallPicInfo() == null ? other.getSmallPicInfo() == null : this.getSmallPicInfo().equals(other.getSmallPicInfo()))

&& (this.getViewPoint() == null ? other.getViewPoint() == null : this.getViewPoint().equals(other.getViewPoint()))

&& (this.getScene() == null ? other.getScene() == null : this.getScene().equals(other.getScene()))

&& (this.getRenderingType() == null ? other.getRenderingType() == null : this.getRenderingType().equals(other.getRenderingType()))

&& (this.getSequence() == null ? other.getSequence() == null : this.getSequence().equals(other.getSequence()))

&& (this.getPanoPath() == null ? other.getPanoPath() == null : this.getPanoPath().equals(other.getPanoPath()))

&& (this.getTaskCreateTime() == null ? other.getTaskCreateTime() == null : this.getTaskCreateTime().equals(other.getTaskCreateTime()))

&& (this.getPid() == null ? other.getPid() == null : this.getPid().equals(other.getPid()))

&& (this.getSysCode() == null ? other.getSysCode() == null : this.getSysCode().equals(other.getSysCode()))

&& (this.getCreator() == null ? other.getCreator() == null : this.getCreator().equals(other.getCreator()))

&& (this.getGmtCreate() == null ? other.getGmtCreate() == null : this.getGmtCreate().equals(other.getGmtCreate()))

&& (this.getModifier() == null ? other.getModifier() == null : this.getModifier().equals(other.getModifier()))

&& (this.getGmtModified() == null ? other.getGmtModified() == null : this.getGmtModified().equals(other.getGmtModified()))

&& (this.getIsDeleted() == null ? other.getIsDeleted() == null : this.getIsDeleted().equals(other.getIsDeleted()))

&& (this.getAtt1() == null ? other.getAtt1() == null : this.getAtt1().equals(other.getAtt1()))

&& (this.getAtt2() == null ? other.getAtt2() == null : this.getAtt2().equals(other.getAtt2()))

&& (this.getNuma1() == null ? other.getNuma1() == null : this.getNuma1().equals(other.getNuma1()))

&& (this.getNuma2() == null ? other.getNuma2() == null : this.getNuma2().equals(other.getNuma2()))

&& (this.getRemark() == null ? other.getRemark() == null : this.getRemark().equals(other.getRemark()))

;

}

@Override

public int hashCode() {

final int prime = 31;

int result = 1;

result = prime \* result + ((getId() == null) ? 0 : getId().hashCode());

result = prime \* result + ((getBusinessId() == null) ? 0 : getBusinessId().hashCode());

result = prime \* result + ((getPicCode() == null) ? 0 : getPicCode().hashCode());

result = prime \* result + ((getPicName() == null) ? 0 : getPicName().hashCode());

result = prime \* result + ((getPicFileName() == null) ? 0 : getPicFileName().hashCode());

result = prime \* result + ((getPicType() == null) ? 0 : getPicType().hashCode());

result = prime \* result + ((getPicSize() == null) ? 0 : getPicSize().hashCode());

result = prime \* result + ((getPicWeight() == null) ? 0 : getPicWeight().hashCode());

result = prime \* result + ((getPicHigh() == null) ? 0 : getPicHigh().hashCode());

result = prime \* result + ((getPicSuffix() == null) ? 0 : getPicSuffix().hashCode());

result = prime \* result + ((getPicLevel() == null) ? 0 : getPicLevel().hashCode());

result = prime \* result + ((getPicFormat() == null) ? 0 : getPicFormat().hashCode());

result = prime \* result + ((getPicPath() == null) ? 0 : getPicPath().hashCode());

result = prime \* result + ((getPicDesc() == null) ? 0 : getPicDesc().hashCode());

result = prime \* result + ((getPicOrdering() == null) ? 0 : getPicOrdering().hashCode());

result = prime \* result + ((getFileKey() == null) ? 0 : getFileKey().hashCode());

result = prime \* result + ((getFileKeys() == null) ? 0 : getFileKeys().hashCode());

result = prime \* result + ((getSmallPicInfo() == null) ? 0 : getSmallPicInfo().hashCode());

result = prime \* result + ((getViewPoint() == null) ? 0 : getViewPoint().hashCode());

result = prime \* result + ((getScene() == null) ? 0 : getScene().hashCode());

result = prime \* result + ((getRenderingType() == null) ? 0 : getRenderingType().hashCode());

result = prime \* result + ((getSequence() == null) ? 0 : getSequence().hashCode());

result = prime \* result + ((getPanoPath() == null) ? 0 : getPanoPath().hashCode());

result = prime \* result + ((getTaskCreateTime() == null) ? 0 : getTaskCreateTime().hashCode());

result = prime \* result + ((getPid() == null) ? 0 : getPid().hashCode());

result = prime \* result + ((getSysCode() == null) ? 0 : getSysCode().hashCode());

result = prime \* result + ((getCreator() == null) ? 0 : getCreator().hashCode());

result = prime \* result + ((getGmtCreate() == null) ? 0 : getGmtCreate().hashCode());

result = prime \* result + ((getModifier() == null) ? 0 : getModifier().hashCode());

result = prime \* result + ((getGmtModified() == null) ? 0 : getGmtModified().hashCode());

result = prime \* result + ((getIsDeleted() == null) ? 0 : getIsDeleted().hashCode());

result = prime \* result + ((getAtt1() == null) ? 0 : getAtt1().hashCode());

result = prime \* result + ((getAtt2() == null) ? 0 : getAtt2().hashCode());

result = prime \* result + ((getNuma1() == null) ? 0 : getNuma1().hashCode());

result = prime \* result + ((getNuma2() == null) ? 0 : getNuma2().hashCode());

result = prime \* result + ((getRemark() == null) ? 0 : getRemark().hashCode());

;

return result;

}

/\*\*获取对象的copy\*\*/

public ResRenderPic copy(){

ResRenderPic obj = new ResRenderPic();

obj.setBusinessId(this.businessId);

obj.setPicCode(this.picCode);

obj.setPicName(this.picName);

obj.setPicFileName(this.picFileName);

obj.setPicType(this.picType);

obj.setPicSize(this.picSize);

obj.setPicWeight(this.picWeight);

obj.setPicHigh(this.picHigh);

obj.setPicSuffix(this.picSuffix);

obj.setPicLevel(this.picLevel);

obj.setPicFormat(this.picFormat);

obj.setPicPath(this.picPath);

obj.setPicDesc(this.picDesc);

obj.setPicOrdering(this.picOrdering);

obj.setFileKey(this.fileKey);

obj.setFileKeys(this.fileKeys);

obj.setSmallPicInfo(this.smallPicInfo);

obj.setViewPoint(this.viewPoint);

obj.setScene(this.scene);

obj.setRenderingType(this.renderingType);

obj.setSequence(this.sequence);

obj.setPanoPath(this.panoPath);

obj.setTaskCreateTime(this.taskCreateTime);

obj.setPid(this.pid);

obj.setSysCode(this.sysCode);

obj.setCreator(this.creator);

obj.setGmtCreate(this.gmtCreate);

obj.setModifier(this.modifier);

obj.setGmtModified(this.gmtModified);

obj.setIsDeleted(this.isDeleted);

obj.setAtt1(this.att1);

obj.setAtt2(this.att2);

obj.setNuma1(this.numa1);

obj.setNuma2(this.numa2);

obj.setRemark(this.remark);

return obj;

}

/\*\*获取对象的map\*\*/

public Map toMap(){

Map map = new HashMap();

map.put("businessId",this.businessId);

map.put("picCode",this.picCode);

map.put("picName",this.picName);

map.put("picFileName",this.picFileName);

map.put("picType",this.picType);

map.put("picSize",this.picSize);

map.put("picWeight",this.picWeight);

map.put("picHigh",this.picHigh);

map.put("picSuffix",this.picSuffix);

map.put("picLevel",this.picLevel);

map.put("picFormat",this.picFormat);

map.put("picPath",this.picPath);

map.put("picDesc",this.picDesc);

map.put("picOrdering",this.picOrdering);

map.put("fileKey",this.fileKey);

map.put("fileKeys",this.fileKeys);

map.put("smallPicInfo",this.smallPicInfo);

map.put("viewPoint",this.viewPoint);

map.put("scene",this.scene);

map.put("renderingType",this.renderingType);

map.put("sequence",this.sequence);

map.put("panoPath",this.panoPath);

map.put("taskCreateTime",this.taskCreateTime);

map.put("pid",this.pid);

map.put("sysCode",this.sysCode);

map.put("creator",this.creator);

map.put("gmtCreate",this.gmtCreate);

map.put("modifier",this.modifier);

map.put("gmtModified",this.gmtModified);

map.put("isDeleted",this.isDeleted);

map.put("att1",this.att1);

map.put("att2",this.att2);

map.put("numa1",this.numa1);

map.put("numa2",this.numa2);

map.put("remark",this.remark);

return map;

}

public String getDesignPlanName() {

return designPlanName;

}

public void setDesignPlanName(String designPlanName) {

this.designPlanName = designPlanName;

}

public String getSpaceType() {

return spaceType;

}

public void setSpaceType(String spaceType) {

this.spaceType = spaceType;

}

public String getArea() {

return area;

}

public void setArea(String area) {

this.area = area;

}

public long getCreateUserId() {

return createUserId;

}

public void setCreateUserId(long createUserId) {

this.createUserId = createUserId;

}

@Override

public String toString() {

return "ResRenderPic{" +

"id=" + id +

", businessId=" + businessId +

", picCode='" + picCode + '\'' +

", picName='" + picName + '\'' +

", picFileName='" + picFileName + '\'' +

", picType='" + picType + '\'' +

", picSize=" + picSize +

", picWeight=" + picWeight +

", picHigh=" + picHigh +

", picSuffix='" + picSuffix + '\'' +

", picLevel=" + picLevel +

", picFormat='" + picFormat + '\'' +

", picPath='" + picPath + '\'' +

", picDesc='" + picDesc + '\'' +

", picOrdering=" + picOrdering +

", fileKey='" + fileKey + '\'' +

", fileKeys='" + fileKeys + '\'' +

", smallPicInfo='" + smallPicInfo + '\'' +

", viewPoint=" + viewPoint +

", scene=" + scene +

", renderingType=" + renderingType +

", sequence=" + sequence +

", panoPath='" + panoPath + '\'' +

", taskCreateTime=" + taskCreateTime +

", pid=" + pid +

", sysCode='" + sysCode + '\'' +

", creator='" + creator + '\'' +

", gmtCreate=" + gmtCreate +

", modifier='" + modifier + '\'' +

", gmtModified=" + gmtModified +

", isDeleted=" + isDeleted +

", att1='" + att1 + '\'' +

", att2='" + att2 + '\'' +

", numa1=" + numa1 +

", numa2=" + numa2 +

", remark='" + remark + '\'' +

", OriginalPicPath='" + OriginalPicPath + '\'' +

", baseRenderId=" + baseRenderId +

", sysTaskPicId=" + sysTaskPicId +

", panoLevel=" + panoLevel +

", roam='" + roam + '\'' +

", designSceneId=" + designSceneId +

", sourcePlanId=" + sourcePlanId +

", templateId=" + templateId +

", fileKeyList=" + fileKeyList +

", renderTaskType='" + renderTaskType + '\'' +

", userId=" + userId +

", productType='" + productType + '\'' +

", totalFee=" + totalFee +

", payType='" + payType + '\'' +

", taskId=" + taskId +

", recommendIds=" + recommendIds +

", sceneIds=" + sceneIds +

", houseId=" + houseId +

", userName='" + userName + '\'' +

", recommendedPlanId=" + recommendedPlanId +

", productReplaceList=" + productReplaceList +

", productDeleteList=" + productDeleteList +

", productGroupReplaceList=" + productGroupReplaceList +

", textureReplaceList=" + textureReplaceList +

", orderNo='" + orderNo + '\'' +

", taskSource=" + taskSource +

", taskType=" + taskType +

", designPlanName='" + designPlanName + '\'' +

", spaceType='" + spaceType + '\'' +

", area='" + area + '\'' +

", planRecommendedId=" + planRecommendedId +

", operationSource=" + operationSource +

", designPlanSceneId=" + designPlanSceneId +

", createUserId=" + createUserId +

'}';

}

public Integer getOverlapResolve()

{

return overlapResolve;

}

public void setOverlapResolve(Integer overlapResolve)

{

this.overlapResolve = overlapResolve;

}

public String getCameraLocation() {

return cameraLocation;

}

public void setCameraLocation(String cameraLocation) {

this.cameraLocation = cameraLocation;

}

}

<!-- 查询所有，可排序 -->

<select id="selectList" resultMap="AllResultMap"

parameterType="com.nork.system.model.ResRenderPic">

select

<include refid="All\_Column\_List" />

from res\_render\_pic

where 1=1

<if test="fileKeyList != null and fileKeyList.size() >0">

and file\_key in

<foreach collection="fileKeyList" index="index" item="item" open="(" separator="," close=")">

#{item}

</foreach>

</if>

<if test="planRecommendedId!= null">and plan\_recommended\_id = #{planRecommendedId,jdbcType=INTEGER} </if>

<if test="designSceneId!= null">and design\_scene\_id = #{designSceneId,jdbcType=INTEGER} </if>

<if test="businessId!= null"> and business\_id = #{businessId,jdbcType=INTEGER} </if>

<if test="picCode!= null"> and pic\_code = #{picCode,jdbcType=VARCHAR} </if>

<if test="picName!= null"> and pic\_name = #{picName,jdbcType=VARCHAR} </if>

<if test="sysTaskPicId!= null"> and sys\_task\_pic\_id = #{sysTaskPicId,jdbcType=INTEGER} </if>

<if test="picFileName!= null"> and pic\_file\_name = #{picFileName,jdbcType=VARCHAR} </if>

<if test="picType!= null"> and pic\_type = #{picType,jdbcType=VARCHAR} </if>

<if test="picSize!= null"> and pic\_size = #{picSize,jdbcType=INTEGER} </if>

<if test="picWeight!= null"> and pic\_weight = #{picWeight,jdbcType=INTEGER} </if>

<if test="picHigh!= null"> and pic\_high = #{picHigh,jdbcType=INTEGER} </if>

<if test="picSuffix!= null"> and pic\_suffix = #{picSuffix,jdbcType=VARCHAR} </if>

<if test="picLevel!= null"> and pic\_level = #{picLevel,jdbcType=INTEGER} </if>

<if test="picFormat!= null"> and pic\_format = #{picFormat,jdbcType=VARCHAR} </if>

<if test="picPath!= null"> and pic\_path = #{picPath,jdbcType=VARCHAR} </if>

<if test="picDesc!= null"> and pic\_desc = #{picDesc,jdbcType=VARCHAR} </if>

<if test="picOrdering!= null"> and pic\_ordering = #{picOrdering,jdbcType=INTEGER} </if>

<if test="fileKey!= null"> and file\_key = #{fileKey,jdbcType=VARCHAR} </if>

<if test="fileKeys!= null"> and file\_keys = #{fileKeys,jdbcType=VARCHAR} </if>

<if test="smallPicInfo!= null"> and small\_pic\_info = #{smallPicInfo,jdbcType=VARCHAR} </if>

<if test="viewPoint!= null"> and view\_point = #{viewPoint,jdbcType=INTEGER} </if>

<if test="scene!= null"> and scene = #{scene,jdbcType=INTEGER} </if>

<if test="renderingType!= null"> and rendering\_type = #{renderingType,jdbcType=INTEGER} </if>

<if test="sequence!= null"> and sequence = #{sequence,jdbcType=INTEGER} </if>

<if test="panoPath!= null"> and pano\_path = #{panoPath,jdbcType=VARCHAR} </if>

<if test="taskCreateTime!= null"> and task\_create\_time = #{taskCreateTime,jdbcType=TIMESTAMP} </if>

<if test="pid!= null"> and pid = #{pid,jdbcType=INTEGER} </if>

<if test="sysCode!= null"> and sys\_code = #{sysCode,jdbcType=VARCHAR} </if>

<if test="creator!= null"> and creator = #{creator,jdbcType=VARCHAR} </if>

<if test="gmtCreate!= null"> and gmt\_create = #{gmtCreate,jdbcType=TIMESTAMP} </if>

<if test="modifier!= null"> and modifier = #{modifier,jdbcType=VARCHAR} </if>

<if test="gmtModified!= null"> and gmt\_modified = #{gmtModified,jdbcType=TIMESTAMP} </if>

<if test="isDeleted!= null"> and is\_deleted = #{isDeleted,jdbcType=INTEGER} </if>

<if test="att1!= null"> and att1 = #{att1,jdbcType=VARCHAR} </if>

<if test="att2!= null"> and att2 = #{att2,jdbcType=VARCHAR} </if>

<if test="numa1!= null"> and numa1 = #{numa1,jdbcType=INTEGER} </if>

<if test="numa2!= null"> and numa2 = #{numa2,jdbcType=INTEGER} </if>

<if test="remark!= null"> and remark = #{remark,jdbcType=VARCHAR} </if>

<if test="sourcePlanId!= null"> and source\_plan\_id = #{sourcePlanId,jdbcType=INTEGER} </if>

<if test="templateId!= null"> and source\_template\_id = #{templateId,jdbcType=INTEGER} </if>

<if test="order != null and order !='' ">

order by ${order}

<if test="orderNum != null and orderNum !='' ">

${orderNum}

</if>

</if>

<if test="orders != null and orders !='' ">

order by ${orders}

</if>

</select>

<!-- \*\*常量定义\*\* -->

<sql id="All\_Column\_List">

id,create\_user\_id,plan\_recommended\_id,business\_id,sys\_task\_pic\_id,pic\_code,pic\_name,pic\_file\_name,pic\_type,pic\_size,

pic\_weight,pic\_high,pic\_suffix,pic\_level,pic\_format,pic\_path,pic\_desc,pic\_ordering,file\_key,file\_keys,small\_pic\_info,

view\_point,scene,rendering\_type,sequence,pano\_path,task\_create\_time,pid,sys\_code,creator,gmt\_create,modifier,gmt\_modified,

is\_deleted,att1,att2,numa1,numa2,remark,pano\_level,roam,source\_template\_id,source\_plan\_id,design\_scene\_id,design\_plan\_name,

space\_type,area,create\_user\_id,plan\_recommended\_id,camera\_location

</sql>

<!-- \*\*结果定义\*\* -->

<resultMap id="AllResultMap" type="com.nork.system.model.ResRenderPic">

<id column="id" property="id" jdbcType="INTEGER" />

<result column="plan\_recommended\_id" property="planRecommendedId" jdbcType="INTEGER" />

<result column="business\_id" property="businessId" jdbcType="INTEGER" />

<result column="pic\_code" property="picCode" jdbcType="VARCHAR" />

<result column="pic\_name" property="picName" jdbcType="VARCHAR" />

<result column="pic\_file\_name" property="picFileName" jdbcType="VARCHAR" />

<result column="pic\_type" property="picType" jdbcType="VARCHAR" />

<result column="pic\_size" property="picSize" jdbcType="INTEGER" />

<result column="pic\_weight" property="picWeight" jdbcType="INTEGER" />

<result column="pic\_high" property="picHigh" jdbcType="INTEGER" />

<result column="pic\_suffix" property="picSuffix" jdbcType="VARCHAR" />

<result column="pic\_level" property="picLevel" jdbcType="INTEGER" />

<result column="pic\_format" property="picFormat" jdbcType="VARCHAR" />

<result column="pic\_path" property="picPath" jdbcType="VARCHAR" />

<result column="pic\_desc" property="picDesc" jdbcType="VARCHAR" />

<result column="pic\_ordering" property="picOrdering" jdbcType="INTEGER" />

<result column="file\_key" property="fileKey" jdbcType="VARCHAR" />

<result column="file\_keys" property="fileKeys" jdbcType="VARCHAR" />

<result column="small\_pic\_info" property="smallPicInfo" jdbcType="VARCHAR" />

<result column="view\_point" property="viewPoint" jdbcType="INTEGER" />

<result column="scene" property="scene" jdbcType="INTEGER" />

<result column="rendering\_type" property="renderingType" jdbcType="INTEGER" />

<result column="sequence" property="sequence" jdbcType="INTEGER" />

<result column="pano\_path" property="panoPath" jdbcType="VARCHAR" />

<result column="task\_create\_time" property="taskCreateTime" jdbcType="TIMESTAMP" />

<result column="pid" property="pid" jdbcType="INTEGER" />

<result column="sys\_code" property="sysCode" jdbcType="VARCHAR" />

<result column="creator" property="creator" jdbcType="VARCHAR" />

<result column="gmt\_create" property="gmtCreate" jdbcType="TIMESTAMP" />

<result column="modifier" property="modifier" jdbcType="VARCHAR" />

<result column="gmt\_modified" property="gmtModified" jdbcType="TIMESTAMP" />

<result column="is\_deleted" property="isDeleted" jdbcType="INTEGER" />

<result column="att1" property="att1" jdbcType="VARCHAR" />

<result column="att2" property="att2" jdbcType="VARCHAR" />

<result column="numa1" property="numa1" jdbcType="INTEGER" />

<result column="numa2" property="numa2" jdbcType="INTEGER" />

<result column="remark" property="remark" jdbcType="VARCHAR" />

<result column="sys\_task\_pic\_id" property="sysTaskPicId" jdbcType="INTEGER" />

<result column="pano\_level" property="panoLevel" jdbcType="INTEGER" />

<result column="roam" property="roam" jdbcType="VARCHAR" />

<result column="design\_scene\_id" property="designSceneId" jdbcType="INTEGER" />

<result column="source\_plan\_id" property="sourcePlanId" jdbcType="INTEGER" />

<result column="source\_template\_id" property="templateId" jdbcType="INTEGER" />

<result column="create\_user\_id" property="createUserId" jdbcType="INTEGER" />

<result column="design\_plan\_name" property="designPlanName" jdbcType="VARCHAR" />

<result column="space\_type" property="spaceType" jdbcType="VARCHAR" />

<result column="area" property="area" jdbcType="VARCHAR" />

<result column="camera\_location" property="cameraLocation" jdbcType="VARCHAR" />

</resultMap>

<!-- \*\*查询定义\*\* -->

<!-- 按主键查询 -->

<select id="selectByPrimaryKey" resultMap="AllResultMap"

parameterType="java.lang.Integer">

select

<include refid="All\_Column\_List" />

from res\_render\_pic

where

<!-- is\_deleted=0 and -->

id = #{id,jdbcType=INTEGER}

</select>

export const sevenCommon = {

data() {

return {

textSwitchList: [],

navOption: {

slidesPerView: 'auto',

slidesPerGroup: 1

},

krpano: document.querySelector('#krpanoSWFObject'),

documentWidth: document.documentElement.clientWidth,

startX: 0,

endX: 0,

moveX: 0,

krpanoNavigationLeft: 0,

cityStoreShow: false,

cityStoreNoResult: false,

preferentialShow: false,

krpanoTechnicalShow: false,

preferentialStoreNoResult: false,

mp3Data: '/static/MP3/music.mp3',

vrListAndProductList: [],

vrListAndProductListObj: {}

count720: 0,

cityStoreList: [],

preferentialObj: {},

isBNextOrpreText: ‘’,

qrCode: false,

qrCodeShow: false,

firstLevelActive: -1,

secondLevelActive: -1,

productsCostTypeList: [],

firstLevelOpen:[],

productDetails: {},

productDetailsBoxShow: false,

goodsListShow: false,

swiperOption: {

slidesPerView: 5,

centeredSlides: false,

spaceBetween: 1,

autoplay: 1500,

// loop: true,

pagination: '.swiper-pagination',

autoplayDisableOnInteraction: false,

simulateTouch: false

},

houseSwiperOption: {

slidesPerView: 'auto',

autoplayDisableOnInteraction: false,

spaceBetween:0

// simulateTouch: false

},

navList: ['全部', '沙发', '休闲椅', '电视柜', '茶几', '角几', '电灯'],

renderPicId: '',

totalPrice: '',

audioTagShow: false,

houseSwiperShow: false,

krpanoNavigationShow: true,

isiphoneOrPC: false,

houseChooseCurrent: 0,

houseChooseList: [],

isFullscreen: false,

routerQueryType: 'seven',

renderSevenList: [],

renderRoamList: [],

freeDesignShow: false,

closeTypeList: ['krpanoRenderListShow','childReplaceBrandBox','childReplaceProductBox', 'krpanoTechnicalShow', 'krpanoReplaceListShow', 'qrCodeShow', 'freeDesignShow', 'houseSwiperShow', 'krpanoExpensesListShow', 'productDetailsBoxShow', 'krpanoAllowReplaceListShow', 'forbidReplaceShow', 'putInMyHouseChooseShow', 'fitmentFinishBoxShow', 'retainOriginalCaseBoxShow'],

krpanoExpensesListShow: false,

goodsDetailsImgActiveIndex: 1,

goodsDetailsImgSwiperOption: {

on:{ slideChangeTransitionEnd(res) { } }

},

conditionSwiperOption: {

spaceBetween: 20,

slidesPerView: 'auto'

},

isReplaceOrWatch: true,

krpanoReplaceListShow: false,

replaceGoodChooseIndex: 0,

aloneGoodsList: [],

replaceIconArr: [],

removeIconArr: [],

krpanoAllowReplaceListShow: false,

messageId: 'recommended', // '任务design'

dataProcessing: ['dians', 'shaf', 'cant', 'chuangt', 'xingx', 'beijing', 'basic\_dians', 'basic\_shaf', 'basic\_cant', 'basic\_chuangt', 'basic\_xingx', 'basic\_beij'],

allowReplaceRroductList: [],

scrollTopFlag: true,

pagesize: 20,

replaceItem: {},

repalceProductTotalCount: 0,

repalceProductIcon: -1,

expensesListType: 'render',

splitReplaceTextureList: [],

splitReplaceTextureActiveArr: [],

krpanoRenderBoxShow: false,

renderTypeList: [],

singleChooseboxActive: 0,

productRenderParams: {

removeParams: [],

aloneReplaceParams: [],

groupReplaceParams: [],

TextureReplaceParams: []

},

promptDialogText: '移除产品渲染后将不会出现在场景中',

promptDialogBoxShow: false,

repalceProductIconItem: {},

operationSource: 1,

isMiniprogram: false,

renderSuccessBoxShow: false,

companyLogo: '',

lackOfIntegralShow: false,

productNoResult: false,

replaceProductNoResult: false,

isShowReplaceIcon: true,

isAloneTextureReplace: true,

isMember: false,

bindingPhoneGetCodeText: '获取验证码',

krpanoBindingMobileBoxShow: false,

bindingPhoneMobileNumber: '',

bindingPhoneMobileCode: '',

MobileCodeAllow: false,

getCodeTimer: null,

proCategory: [],

productTypeArr: [],

childProductList: [],

productTypeActive: 0,

smallCategoryCode: '',

childReplaceProductBox: false,

childProductActiveArr: [],

dataProcessing: ['dians', 'shaf', 'cant', 'chuangt', 'xingx', 'beijing', 'basic\_dians', 'basic\_shaf', 'basic\_cant', 'basic\_chuangt', 'basic\_xingx', 'basic\_beij'],

productSmallTypeValue: '',

childProductListTitleActive: [],

moreBrandList: [],

moreBrandSortList: {

A: [],

B: [],

C: [],

D: [],

E: [],

F: [],

G: [],

H: [],

I: [],

J: [],

K: [],

L: [],

M: [],

N: [],

O: [],

P: [],

Q: [],

R: [],

S: [],

T: [],

U: [],

V: [],

W: [],

X: [],

Y: [],

Z: [],

'#': []

},

childReplaceBrandBox: false,

brandContentSecondIndex: -1,

chnNumChar: ['零','一', '二', '三', '四','五', '六', '七', '八', '九'],

krpanoRenderListShow: false,

renderInventoryList: [],

mobileToBUrl: '',

isShowPayAmountPop: false,

isShowPayAmountStylePopBox: false,

isShowPaySuccessPopBox: false,

payAmount: '',

isShowPayFailPopBox: false,

payType: 0,

isRenderOrNoDubi: 'render',

isRender: '-1',

planId: '',

forbidReplaceShow: false,

promptBoxText: "此方案的其它空间正在渲染中，待渲染完成后才能产品替换哦~",

fitmentFinishBoxShow: false,

isAllHouseOrAloneSpace: 0,

renderState: 1,

fullHousePlanAction: 3,

retainOriginalCaseBoxShow: false,

isTask: 0,

taskType: 1,

putInMyHouseChooseShow: false,

isShowReplace: 0,

miniProgramPostMessage: {

renderIdParams: {},

shareText: '',

atPresentUuid: ''

}

}

},

created() {

this.isMiniprogramFn()

this.getRouterParams()

this.phoneOrPcDispose()

},

computed: {

...mapGetters(['token', 'resourceUrl', 'loadingShow', 'mobile2bUrl']),

renderBtnActive() {

return Object.keys(this.productRenderParams).filter(value => this.productRenderParams[value].length > 0).length > 0

},

krpanoCommonMaskShow() {

return this.closeTypeList.filter(value => { return (this[value] && value !== 'houseSwiperShow')}).length > 0

},

bindingPhoneAllow() {

let regMobile = /(^1[3|4|5|6|7|8|9]\d{9}$)|(^09\d{8}$)/, regCode = /^\d{6}\b/

let flagMobile = regMobile.test(this.bindingPhoneMobileNumber), flagCode = regCode.test(this.bindingPhoneMobileCode)

return flagMobile && flagCode

},

MobileNumberAllow() {

let regMobile = /(^1[3|4|5|6|7|8|9]\d{9}$)|(^09\d{8}$)/

let flag = regMobile.test(this.bindingPhoneMobileNumber)

return flag

}

},

watch: {

childProductList(newVal) {

this.childProductListTitleActive = newVal.map(() => false)

},

'$route'(newRouter) {

if (newRouter.query.payIsSuccess == 1) {

this.isShowPaySuccessPopBox = !(this.isShowPayAmountPop = this.isShowPayFailPopBox = false)

} else if (newRouter.query.payIsSuccess == 2){

this.isShowPayFailPopBox = !(this.isShowPayAmountPop = false)

}

}

},

methods: {

...mapActions(['setToken', 'setPlanId', 'setRenderPicId']),

processorUrlList(url, isShear) {

return isShear == 1 ? `

<preview url="${url+ '/preview.jpg'}" />

<image>

<cube url="${url + '/pano\_%s.jpg'}" />

</image>

` : `<image><sphere url="${url}" /></image>`

},

onReadyTest() {

let name = `<scene name="test" title="1" onstart="" havevrimage="true" thumburl="/static/panos/1.tiles/thumb.jpg" lat="" lng="" heading="">

<preview url="/static/pano/preview.jpg"/>

<image><cube url="/static/pano/pano\_%s.jpg" /></image>

<autorotate enabled="true" waittime="4" accel="1.0" speed="5.0" horizon="0.0"/>

<view stereographic="true" fisheye="1.0" fov="155" fovtype="VFOV" fovmin='80' limitview="auto" fovmax="120" hlookat="-60" vlookat="77"/>

</scene>

`

this.krpano.call('loadxml(' + name + ');loadscene('+'test'+')');

},

watchMobilePostMessage() {

window.addEventListener('message', (e) => {

e.data == 'paySuccess' ? this.isShowPaySuccessPopBox = !(this.isShowPayAmountStylePopBox = this.isShowPayFailPopBox = false) :

(e.data == 'payFail' ? this.isShowPayFailPopBox = !(this.isShowPayAmountStylePopBox = false) : null)

})

},

navigationSlideStart(event) {

this.documentWidth = document.documentElement.clientWidth

if (this.documentWidth >= 768) { return }

this.startX = event.touches[0].pageX

this.krpanoNavigationLeft = this.$refs.krpanoNavigation.offsetLeft

this.krpanoNavigationRight = this.$refs.krpanoNavigation.offsetRight

},

navigationSlideMove(event) {

this.moveX = event.touches[0].pageX

if (this.moveX - this.startX > 0 && this.$refs.krpanoNavigation.getBoundingClientRect().left >= 26) {

this.$refs.krpanoNavigation.style.left = '26px'

return

}

else if (this.moveX - this.startX <= 0 && this.$refs.krpanoNavigation.getBoundingClientRect().right <= 86){

return

}

this.$refs.krpanoNavigation.style.left = `${this.krpanoNavigationLeft + (this.moveX - this.startX)}px`

},

navigationSlideEnd(event) {

this.startX = 0

this.moveX = 0

},

navigationFunction(param) {

if (param !== 'music') { this.productDetailsBoxShow = false }

switch(param) {

case 'VR' :this.openVR();break;

case 'music' : this.audioIsPlay(); break;

case 'sharing' : this.getQrcode(); break;

case 'goods' : this.getProductDetailsList(); break;

case 'houseChoose': this.houseSwiperListShow(); break;

case 'screen' : this.requestFullScreen(); break;

case 'freeDesign' : this.AsRequiredClose('freeDesignShow'); break;

case 'goodsList' : this.AsRequiredClose('krpanoExpensesListShow'); this.getGoodsList('list'); break;

case 'replace' : this.isAllowReplace(); break;

case 'technical' : this.AsRequiredClose('krpanoTechnicalShow'); break;

case 'style' : this.AsRequiredClose('krpanoSwitchStyleShow');this.getRelevanceCaseCondition(); break;

case 'store' : this.AsRequiredClose('krpanoGoodsStoreShow'); this.getCityStore(); break;

case 'activity' : this.AsRequiredClose('krpanoDiscountsActivityShow'); this.getCityActivity(); break;

case 'renovationMyHouse': this.renovationMyHouse()

}

},

isAllowReplace() {

if (this.isRender === '3') {

if ( this.renderState == 2 ) { return (this.forbidReplaceShow = true, this.promptBoxText = '此方案的其它空间正在渲染中，待渲染完成后才能产品替换哦~') }

}

this.AsRequiredClose('krpanoReplaceListShow')

this.getGoodsList('replace')

},

openVR() {

this.$ba.trackEvent('Virtual Reality', 'click')

this.krpano.call('WebVR.enterVR();')

},

audioIsPlay() { // music

this.$ba.trackEvent('Sound Settings', 'click')

this.$refs.audioTag !== null ? this.$refs.audioTag[(this.audioTagShow = this.$refs.audioTag.paused) ? 'play' : 'pause']() : null

},

closeAllTypeList(whoShow) { // 关闭所有弹框

this.utils.myForEach(this.closeTypeList, (value) => { this[value] = false })

if (whoShow) { // 显示某个

this[whoShow === 'common' ? (this.isReplaceOrWatch ? 'krpanoExpensesListShow' : (this.isAloneTextureReplace ? 'krpanoReplaceListShow': 'krpanoAllowReplaceListShow')) : whoShow] = true

}

},

firstLevelActivechange(indexP) { // 改变第一级样式

this.$set(this.firstLevelOpen[indexP], 'active', !this.firstLevelOpen[indexP].active)

},

secondLevelActivechange(indexP, indexC) {

this.$set(this.firstLevelOpen[indexP].dataListActive, indexC, !this.firstLevelOpen[indexP].dataListActive[indexC])

},

houseSwiperListShow() {

this.$ba.trackEvent('Plan Change', 'click')

this.AsRequiredClose('houseSwiperShow')

this.krpanoNavigationShow = false

},

chooseHouse(index) {

this.houseChooseCurrent = index

let urlList = this.routerQueryType === 'roam' ? ((this.roam720UrlList = this.houseChooseList[index].singleSceneVo.roamList) ? this.roam720UrlList[0] : this.houseChooseList[index].singleSceneVo) :

this.houseChooseList[index].singleSceneVo

this.companyLogo = this.houseChooseList[index].singleSceneVo.panoramaVo.logoPath

this.onReady(urlList, 'start')

this.krpanoNavigationShow = true

},

backTop() {

this.$refs.krpanoGoodsList.scrollTop = 0

},

detailsbackTop() { // 返回详情顶部

this.$refs.krpanoProductDetails.scrollTop = 0

},

requestFullScreen() {

this.$ba.trackEvent('Full-Screen Preview', 'click')

let ele = document.documentElement

if (ele.requestFullScreen) {

ele.requestFullScreen()

} else if (ele.webkitRequestFullScreen) {

ele.webkitRequestFullScreen()

} else if (ele.mozRequestFullScreen) {

ele.mozRequestFullScreen ()

} else if (ele.msRequestFullScreen) {

ele.msRequestFullScreen()

}

this.isFullscreen = true

},

exitScreen() {

let ele = document

if (ele.exitFullscreen) {

ele.exitFullscreen()

} else if (ele.webkitCancelFullScreen) {

ele.webkitCancelFullScreen()

} else if (ele.mozCancelFullScreen) {

ele.mozCancelFullScreen()

} else if (ele.msExitFullscreen) {

ele.msExitFullscreen()

}

this.isFullscreen = false

},

AsRequiredClose(type) {

this.utils.myForEach(this.closeTypeList, (value) => {

value === type? this[type] = !this[type] : this[value] = false

})

if (!this.houseSwiperShow) { this.krpanoNavigationShow = !this.houseSwiperShow }

},

isHardOutfit(result) {

let wallCategoryCode = ['dians', 'shaf', 'cant', 'chuangt', 'beijing', 'mengk']

return result.filter((value) => {

return value.detailList.filter((val) => val.detailList = val.detailList.map((item) => Object.assign(item, { flag: value.costTypeName === '硬装' ? (wallCategoryCode.findIndex((n) => item.smallCategoryCode === n) === -1) : false})))

})

},

filterGoodsList(detailList) {

return detailList.reduce((arr, item) => { return arr.concat(item.detailList)}, [])

},

chooseCostType(index) {

this.aloneGoodsList = this.filterGoodsList(this.productsCostTypeList[index].detailList)

this.replaceGoodChooseIndex = index

},

removeProduct(index, item) {

if (item.flag) { this.showPromptDialogBox('硬装不可移除'); return }

if (this.removeIconArr[this.replaceGoodChooseIndex][index] != -1 || this.removeIconArr[this.replaceGoodChooseIndex][index] != -1) { return }

this.productRenderParams.removeParams.push({ sourceProductId: item.productId, sourceProductCode: item.productCode })

this.showPromptDialogBox('移除产品渲染后将不会出现在场景中')

this.$set(this.removeIconArr[this.replaceGoodChooseIndex], index, index)

},

getChildProductActiveArr(list) {

return list instanceof Array ? list.map(() => -1) : []

},

changeChildProductActiveArr(parantIndex, childIndex, type) {

let reg = /^[A-Z]+$/

if (type === 'moreBrand') {

this.utils.myForEach(this.moreBrandList, (value, index) => {序

value.parantIndex = parantIndex

value.childIndex = index

if (reg.test(this.pySort(value.categoryName)[0].toUpperCase())) {

this.moreBrandSortList[this.pySort(value.categoryName)[0].toUpperCase()].push(value)

} else {

this.moreBrandSortList['#'].push(value)

}

})

this.AsRequiredClose('childReplaceBrandBox')

return

}

this.$set(this.childProductActiveArr, parantIndex, this.childProductActiveArr[parantIndex] === childIndex? -1 : childIndex)

},

replaceProduct(index, item) {

this.replaceItem = item

this.replaceItemIndex = index

this.productTypeActive = 0

this.theDataRegression()

this.AsRequiredClose('krpanoAllowReplaceListShow')

this.proCategoryLongCode = item.proCategoryLongCode

this.productSmallTypeValue = item.productSmallTypeValue

this.smallCategoryCode = this.errorDataFiter(item.smallCategoryCode)

this.productTypeArr = this.childProductList = [];

// this.childProductList = this.proCategoryFilter({type: 'child', index: 0})

// this.childProductActiveArr = this.getChildProductActiveArr(this.childProductList)

this[item.isMainProduct === 1? 'searchGroupProductApi' : 'searchAloneProductApi'](item)

},

resetChildReplaceProductBox() {

this.childProductActiveArr = this.getChildProductActiveArr(this.childProductList)

},

colseChildReplaceProductBox() {

this.AsRequiredClose('krpanoAllowReplaceListShow')

this.childProductActiveArr = this.getChildProductActiveArr(this.childProductList)

},

colseChildReplaceBrandBox() {

Object.keys(this.moreBrandSortList).forEach(value => { this.moreBrandSortList[value] = []})

this.brandContentSecondIndex = -1

this.AsRequiredClose('childReplaceProductBox')

},

confirmChildReplaceBrandChoose() {

if (this.brandContentSecondIndex !== -1) {

let item = this.moreBrandSortList[this.brandContentSecondIndex[0]][this.brandContentSecondIndex.slice(1)]

this.$set(this.childProductActiveArr, item.parantIndex, item.childIndex)

this.brandContentSecondIndex = -1

}

Object.keys(this.moreBrandSortList).forEach(value => { this.moreBrandSortList[value] = []})

this.AsRequiredClose('childReplaceProductBox')

},

chooseReplaceFiltrateBrand(key, index) {

this.brandContentSecondIndex = key + index

},

flitrateReplaceProduct() {

let codeStr = []

this.utils.myForEach(this.childProductActiveArr, (value, index) => {

if (value !== -1) {

let code = this.childProductList[index].productCategoryVoList[value].categoryCode

codeStr.push(code.slice(1, code.length - 1))

}

})

if (codeStr.length > 0) {

this.childProductActiveArr = this.getChildProductActiveArr(this.childProductList)

this.AsRequiredClose('krpanoAllowReplaceListShow')

this.childProductActiveArr = []

this.proCategoryLongCode = codeStr.join(',')

this.theDataRegression()

this.replaceItem.isMainProduct === 1? this.searchGroupProductApi(this.replaceItem) : this.searchAloneProductApi(this.replaceItem, 'noAllow')

}

},

theDataRegression() {

this.allowReplaceRroductList = []

this.repalceProductTotalCount = 0

this.repalceProductIcon = -1

this.pagesize = 20

this.repalceProductTotalCount = 0

this.replaceProductNoResult = false

},

errorDataFiter(smallCategoryCode) {

let obj = {

mengk: 'mengk',

chuangk: 'mengk',

itho: 'it00',

jiem: 'zaox'

}

return this.dataProcessing.indexOf(smallCategoryCode) !== -1 ? 'xingx' : (obj[smallCategoryCode] ? obj[smallCategoryCode] : smallCategoryCode)

},

proCategoryFilter(options) {

return options.type ? this.productTypeArr[this.productTypeActive = options.index].ff.filter((value) => value.dd !== '品牌') :

this.proCategory

.reduce((arr, value) => { value.bb === options.categoryCode ? arr.push(value.ff) : null; return arr}, [])

.filter(value => { value.bb === options.smallCategoryCode })

},

proCategorySwitch(item, index) {

this.productTypeActive = index

this.childProductList = this.filterChildProductList(this.productTypeArr[index].productCategoryVoList)

this.childProductActiveArr = this.getChildProductActiveArr(this.childProductList)

this.proCategoryLongCode = item.categoryCode

this.productSmallTypeValue = item.productSmallTypeValue

this.theDataRegression()

if (this.replaceItem.isMainProduct === 1) {

this.searchGroupProductApi(this.replaceItem)

} else {

this.searchAloneProductApi(this.replaceItem, 'noAllow')

}

},

searchGroupProductApi(item) {

let params = {}

params = {

limit: this.pagesize,

start: 0,

categoryCode: this.smallCategoryCode,

houseType: item.houseType,

userId: 582,

planProductId: item.planProductId,

spaceCommonId: item.spaceCommonId,

productTypeValue: item.productTypeValue,

smallTypeValue: item.productSmallTypeValue,

msgId: this.messageId,

productId: item.productId

}

this.API.getProductschangeList(params)

.then((res) => {

// type === 'noAllow'?'' : this.AsRequiredClose('krpanoAllowReplaceListShow')

if (res.success) {

this.scrollTopFlag = true

if (res.datalist && res.datalist.length > 0) {

this.allowReplaceRroductList = res.datalist

this.repalceProductTotalCount = res.totalCount

this.replaceProductNoResult = false

} else {

this.replaceProductNoResult = true

this.allowReplaceRroductList = []

this.repalceProductTotalCount = 0

}

} else {

this.replaceProductNoResult = true

this.allowReplaceRroductList = []

this.repalceProductTotalCount = 0

}

}).catch(err => {

// type === 'noAllow'?'' : this.AsRequiredClose('krpanoAllowReplaceListShow')

this.replaceProductNoResult = true

this.allowReplaceRroductList = []

this.repalceProductTotalCount = 0

})

},

searchAloneProductApi(item, type) {

let params = {}

let designPlanType = this.messageId === 'recommended' ? 2 : 3

params = {

productTypeValue: item.productTypeValue,

productTypeSmallValue: this.productSmallTypeValue,

start: 0,

dataSize: this.pagesize,

productCategoryLongCode: this.proCategoryLongCode,

designPlanProductId: item.planProductId,

designPlanType: designPlanType,

productId: item.productId,

productBrandId: item.brandId

}

if (type === 'noAllow') {

this.secondSearchAloneProductApi(params)

} else {

this.firstSearchAloneProductApi(params)

}

},

async firstSearchAloneProductApi(params) {

let $self = this

function setEmpty() {

$self.replaceProductNoResult = true

$self.allowReplaceRroductList = []

$self.repalceProductTotalCount = 0

}

let res = await this.API.searchAloneProductList(Object.assign(params, { aggsData: 'Y', productTypeSmallValue: this.productSmallTypeValue}))

if (res.success) {

if (res.datalist) {

this.productTypeArr = res.datalist.aggs

this.productCategoryLongCode = this.productTypeArr[0].categoryCode

this.productSmallTypeValue = this.productTypeArr[0].productSmallTypeValue

let resSecond = await this.API.searchAloneProductList(Object.assign(params, { aggsData: 'N', productTypeSmallValue: this.productSmallTypeValue, bizProductTypeSmallValue: this.productSmallTypeValue}))

if (resSecond.datalist) {

this.utils.myForEach(resSecond.datalist.obj, (value) => {

value.brandName = value.productBrandName

value.unitPrice = value.productSalePrice

value.productId = value.id

})

this.allowReplaceRroductList = resSecond.datalist.obj

this.childProductList = this.filterChildProductList(res.datalist.aggs[0].productCategoryVoList)

this.childProductActiveArr = this.getChildProductActiveArr(this.childProductList)

this.repalceProductTotalCount = resSecond.totalCount

this.scrollTopFlag = true

this.replaceProductNoResult = false

} else {

setEmpty()

}

} else {

setEmpty()

}

} else {

setEmpty()

}

},

secondSearchAloneProductApi(params) {

this.API.searchAloneProductList(Object.assign(params, { aggsData: 'N', productTypeSmallValue: this.productSmallTypeValue, bizProductTypeSmallValue: this.productSmallTypeValue}))

.then((res) => {

// type === 'noAllow'?'' : this.AsRequiredClose('krpanoAllowReplaceListShow')

if (res.success) {

if (res.datalist) {

this.utils.myForEach(res.datalist.obj, (value) => {

value.brandName = value.productBrandName

value.unitPrice = value.productSalePrice

value.productId = value.id

})

this.allowReplaceRroductList = res.datalist.obj

this.repalceProductTotalCount = res.totalCount

this.scrollTopFlag = true

this.replaceProductNoResult = false

} else {

this.replaceProductNoResult = true

this.allowReplaceRroductList = []

this.repalceProductTotalCount = 0

}

} else {

this.replaceProductNoResult = true

this.allowReplaceRroductList = []

this.repalceProductTotalCount = 0

}

}).catch((res) => {

// type === 'noAllow'?'' : this.AsRequiredClose('krpanoAllowReplaceListShow')

this.replaceProductNoResult = true

this.allowReplaceRroductList = []

this.repalceProductTotalCount = 0

})

},

filterChildProductList(childProductList) {

this.utils.myForEach(childProductList, (value) => {

value.productCategoryVoList.sort(this.compare('categoryOrder'))

if (value.categoryName == '品牌' && value.productCategoryVoList.length > 12) {

this.moreBrandList = value.productCategoryVoList

}

})

return childProductList.sort(this.compare('categoryOrder'))

},

compare(prop) {

return function (obj1, obj2) {

return Math.sign(obj1[prop] - obj2[prop])

}

},

recoverProduct(index,item, type) {

let replaceGoodChooseIndex = type === 'renderInventory' ? item.indexP : this.replaceGoodChooseIndex

if (this.removeIconArr[replaceGoodChooseIndex][index] !== -1) {

let index = this.findIndex(this.productRenderParams.removeParams, 'sourceProductId', item.productId)

this.productRenderParams.removeParams.splice(index, 1)

} else if (this.replaceIconArr[replaceGoodChooseIndex][index] !== -1){

if (item.isMainProduct === 0) {

let index = this.findIndex(this.productRenderParams.aloneReplaceParams, 'sourceProductId', item.productId)

this.productRenderParams.aloneReplaceParams.splice(index, 1)

} else {

let index = this.findIndex( this.productRenderParams.groupReplaceParams, 'sourceProductId', item.productId)

this.productRenderParams.groupReplaceParams.splice(index, 1)

}

let index = this.findIndex(this.productRenderParams.TextureReplaceParams, 'posIndexPath' ,item.posIndexPath)

index !== undefined ? this.productRenderParams.TextureReplaceParams.splice(index, 1) : null

}

this.$set(this.removeIconArr[replaceGoodChooseIndex], index, -1)

this.$set(this.replaceIconArr[replaceGoodChooseIndex], index, -1)

type === 'renderInventory' ? this.setKrpanoRenderListShow('recover'):''

this.showPromptDialogBox('已成功恢复原产品')

},

findIndex(paramsArr, key, id) {

return paramsArr.findIndex((value) => { return value[key] === id })

},

scrollBottom(ref) {

if (this.allowReplaceRroductList.length >= this.repalceProductTotalCount) { return }

let [sh, st, h] = [(this.$refs[ref].scrollHeight - 2), this.$refs[ref].scrollTop, this.$refs[ref].offsetHeight]

if (this.scrollTopFlag) {

if (st + h >= sh) {

this.scrollTopFlag = false

this[this.replaceItem.isMainProduct === 1? 'searchGroupProductApi' : 'searchAloneProductApi'](this.replaceItem, 'noAllow')

this.pagesize += 20

}

}

},

chooseReplaceProduct(index) {

this.repalceProductIcon = this.repalceProductIcon === index ? -1: index

},

getReplaceAllProductDetails(item) {

this.isAloneTextureReplace = false

this.getReplaceProductDetails(item)

},

getReplaceAloneProductDetails(item, index){

this.replaceItemIndex = index

this.replaceItem = item

this.isAloneTextureReplace = true

this.getReplaceProductDetails(item)

},

getReplaceProductDetails(item) {

this.isReplaceOrWatch = false

this.repalceProductIconItem = item

let productId = item.productId

this.splitReplaceTextureActiveArr = []

this.API.getGoodDetails({ id: productId })

.then((res) => {

if (res.status) {

if (res.obj) {

this.productDetails = res.obj

this.goodsDetailsImgActiveIndex = 1

this.AsRequiredClose('productDetailsBoxShow')

if (res.obj.splitTextureList) {

this.splitReplaceTextureList = res.obj.splitTextureList

this.utils.myForEach(this.splitReplaceTextureList, (valP) => {

this.splitReplaceTextureActiveArr.push(0)

})

} else {

this.splitReplaceTextureList = []

}

} else {

this.productDetails = {}

this.splitReplaceTextureList = []

}

} else {

this.productDetails = {}

this.splitReplaceTextureList = []

}

})

},

switchTextureActive(indexP, indexC) {

this.$set(this.splitReplaceTextureActiveArr, indexP, indexC)

},

addReplaceRenderTask() {

if (this.repalceProductIcon === -1) { return }

let item = this.allowReplaceRroductList[this.repalceProductIcon]

if (this.replaceItem.isMainProduct === 0) {

this.API.getDetailByIds({

idArrs: item.productId.toString(),

limit: 1,

start: 0,

userId: 0,

isStandard: this.replaceItem.isStandard,

regionMark: this.replaceItem.regionMark,

styleId: this.replaceItem.styleId,

measureCode: this.replaceItem.measureCode,

designPlanId: this.replaceItem.designPlanId,

planProductId: this.replaceItem.planProductId,

msgId: this.messageId

}).then((res) => {

this.closeAllTypeList('krpanoReplaceListShow')

if (res.success && res.datalist) {

item = res.datalist[0]

this.$set(this.replaceIconArr[this.replaceGoodChooseIndex], this.replaceItemIndex, this.replaceItemIndex)

this.productRenderParams.aloneReplaceParams.push({

destProductCode: item.productCode,

destProductId: item.productId,

posIndexPath: this.replaceItem.posIndexPath,

sourceProductCode: this.replaceItem.productCode,

sourceProductId: this.replaceItem.productId

})

}

})

} else {

let planId = this.replaceItem.productCount > 1?this.planId : 0

this.productRenderParams.groupReplaceParams.push({

planId: planId,

replaceGroupCode: item.groupCode,

replaceGroupId: item.groupId,

sourceGroupProductCode: this.replaceItem.sourceGroupProductCode,

sourceGroupProductId: this.replaceItem.sourceGroupProductId,

sourceGroupProductUniqueId: this.replaceItem.sourceGroupProductUniqueId,

sourcePlanGroupId: this.replaceItem.sourcePlanGroupId,

sourcePosIndexPath: this.replaceItem.posIndexPath,

sourceProductGroupId: this.replaceItem.sourceProductGroupId,

sourceProductId: this.replaceItem.productId

})

this.closeAllTypeList('krpanoReplaceListShow')

this.$set(this.replaceIconArr[this.replaceGoodChooseIndex], this.replaceItemIndex, this.replaceItemIndex)

}

},

fromDetailsAddRender(item) {

if (this.splitReplaceTextureList.length === 0 && this.isAloneTextureReplace) {

this.showPromptDialogBox('无材质可换')

return

}

let materialString = ''

let COUNT\_ = 0

let replaceItem = {}

this.utils.myForEach(this.splitReplaceTextureActiveArr, (value, index) => {

if (value !== 0) {

COUNT\_++

materialString += `${COUNT\_}:${this.splitReplaceTextureList[index].textureList[value].id};`

}

})

if (materialString) {

materialString = materialString.slice(0, materialString.length - 1)

this.productRenderParams.TextureReplaceParams.push({

posIndexPath: this.replaceItem.posIndexPath,

replaceSplitTexturesParam: materialString,

sourcePlanProductId: this.planId,

sourceProductId: item.productId,

sourceSplitTexturesChooseInfo: item.splitTexturesInfo

})

}

if (this.isAloneTextureReplace) {

if (materialString) {

this.$set(this.replaceIconArr[this.replaceGoodChooseIndex], this.replaceItemIndex, this.replaceItemIndex)

this.closeAllTypeList('krpanoReplaceListShow')

} else {

this.showPromptDialogBox('请选择替换材质')

}

return

}

if (this.replaceItem.isMainProduct === 0) {

this.API.getDetailByIds({

idArrs: item.productId.toString(),

limit: 1,

start: 0,

userId: 0,

isStandard: this.replaceItem.isStandard,

regionMark: this.replaceItem.regionMark,

styleId: this.replaceItem.styleId,

measureCode: this.replaceItem.measureCode,

designPlanId: this.replaceItem.designPlanId,

planProductId: this.replaceItem.planProductId,

msgId: this.messageId

})

.then((res) => {

this.closeAllTypeList('krpanoReplaceListShow')

if (res.success && res.datalist) {

replaceItem = res.datalist[0]

this.$set(this.replaceIconArr[this.replaceGoodChooseIndex], this.replaceItemIndex, this.replaceItemIndex)

this.productRenderParams.aloneReplaceParams.push({

destProductCode: replaceItem.productCode,

destProductId: replaceItem.productId,

posIndexPath: this.replaceItem.posIndexPath,

sourceProductCode: this.replaceItem.productCode,

sourceProductId: this.replaceItem.productId

})

}

})

} else {

let planId = this.replaceItem.productCount > 1 ? this.planId : 0

this.productRenderParams.groupReplaceParams.push({

planId: planId,

replaceGroupCode: this.repalceProductIconItem.groupCode,

replaceGroupId: this.repalceProductIconItem.groupId,

sourceGroupProductCode: this.replaceItem.sourceGroupProductCode,

sourceGroupProductId: this.replaceItem.sourceGroupProductId,

sourceGroupProductUniqueId: this.replaceItem.sourceGroupProductUniqueId,

sourcePlanGroupId: this.replaceItem.sourcePlanGroupId,

sourcePosIndexPath: this.replaceItem.posIndexPath,

sourceProductGroupId: this.replaceItem.sourceProductGroupId,

sourceProductId: this.replaceItem.productId

})

this.closeAllTypeList('krpanoReplaceListShow')

this.$set(this.replaceIconArr[this.replaceGoodChooseIndex], this.replaceItemIndex, this.replaceItemIndex)

}

},

getRenderType() {

this.API.getRenderType()

.then((res) => {

this.renderTypeList = res.status ? res.obj : []

})

},

showRenderTypeBox () {

this.payAmount = ''

this.isShowPaySuccessPopBox = false

if (!this.renderBtnActive) {

this.showPromptDialogBox('无渲染数据')

return

}

// this.isRender === '3' || this.taskType === '3' ? this.retainOriginalCaseBoxShow = true : this.confirmBeginRender()

this.API.checkReplaceDesignPlanPay(this.isAllHouseOrAloneSpace === 0 ? { [this.operationSource == 1 ? 'recommendedPlanId' : 'designPlanRenderSceneId']: this.planId, planType: 1} : { [this.operationSource == 1 ? 'recommendedPlanId' : 'designPlanRenderSceneId']: this.putInMyHouseRenderParams.planRecommendedId, planType: 0}).then(res => {

if (res.obj.flag) { // 只有查看效果图，跟我的方案里面有户型的方案，才会提示新增或者替换

this.isRender === '3' || this.taskType === '3' ? this.retainOriginalCaseBoxShow = true : this.confirmBeginRender()

} else {

this.payDubiBoxShow = true

}

})

},

selectPayType(index) {

this.payType = index

},

payDesignPlan(){

let postPayData = { planRecommendedId: this.planId, useType: 1 }

this.isMiniprogram ? Object.assign(postPayData, {

payMethod: 'miniProPay',

payType: 1

}) : Object.assign(postPayData, {

payMethod: this.payType ? 'aliAppPay' : 'wxAppPay',

payType: this.payType

})

this.API.payDesignPlanCopyRight(postPayData).then(res => {

if(this.isMiniprogram){

wx.miniProgram.navigateTo({ url: `/pages/pay/pay?timestamp=${res.obj.timeStamp}&nonceStr=${res.obj.nonceStr}&signType=${res.obj.signType}&paySign=${res.obj.paySign}&prepayId=${res.obj.package.slice(10)}&appId=${res.obj.appId}` })

this.isShowPayAmountPop = false

} else if (storage.session.get('platformCode') === 'mobile2b' && storage.session.get('token') && this.isShowReplaceIcon) {

parent.postMessage(`plan=${this.planId}&type=${this.payType}`, this.mobileToBUrl)

} else {

this.showPromptDialogBox('该版本不支持支付')

}

})

},

confirmBeginRender() {

let item = this.renderTypeList[this.singleChooseboxActive]

let renderValue = { 1: 'common\_render', 4: 'panorama\_render', 6: 'video', 8: 'roam720' }[item.renderValue]

let renderingParameter = {

renderTaskType: renderValue,

operationSource: this.operationSource, // 入口为推荐方案则传1，入口为我的方案传0

houseId: 0,

productDeleteList: this.productRenderParams.removeParams, // 产品移除

productGroupReplaceList: this.productRenderParams.groupReplaceParams, // 组合产品替换数组

productReplaceList: this.productRenderParams.aloneReplaceParams, // 单品替换数组

textureReplaceList: this.productRenderParams.TextureReplaceParams, // 材质替换

taskSource: 0, // web传0，移动端传1

taskType: 1, // 渲染任务传1，装进我家传0

totalFee: this.isRender != '-1' ? 0 : (this.isMember ? 0 : 1) // 是否需要支付度币

}

this.API.addRenderTask(this.isAllHouseOrAloneSpaceParams(renderingParameter))

.then((res) => {

this.krpanoRenderBoxShow = false

if (res.success) {

this.miniProgramPostMessage.renderIdParams = res.obj

this.isMiniprogram ? wx.miniProgram.postMessage({ data: this.miniProgramPostMessage }) : null // 渲染成功，触发回调

this.dataRegressionApi(item.renderValue)

} else {

let text = res.message === '请登录!' ? '登录超时，请重新登录' : (res.message === '账户度币不足，请充值.' || res.message === '用户度币信息为空，生成订单失败' ? '账户度币不足，请充值' : '渲染失败')

res.message === '账户度币不足，请充值.' || res.message === '用户度币信息为空，生成订单失败' ? this.lackOfIntegralShow = true : null

this.showPromptDialogBox(text)

}

})

},

isAllHouseOrAloneSpaceParams(renderingParameter) {

this.isAllHouseOrAloneSpace === 1 ? Object.assign(renderingParameter, { // 是否是全屋

planHouseType: 3,

fullHousePlanAction: this.fullHousePlanAction,

preRenderSceneId: this.planId,

spaceFunctionId: this.designPlanType

}, this.productReplaceParams) : null

this.operationSource === 1 ? Object.assign(renderingParameter, {

planRecommendedId: this.planId,

designPlanSceneId: null

}) : Object.assign(renderingParameter, {

planRecommendedId: null,

designPlanSceneId: this.planId

})

return renderingParameter

},

dataRegressionApi(renderValue) {

if (this.templateFormId) {

this.API.pushTemplateMessage({

designPlanId: this.planId,

formId: this.templateFormId,

forwardPage:'pages/my-tasks/my-tasks',

renderType: 1,

renderLevel: renderValue

})

.then(res => {

this.templateFormId = ''

console.log(res, '模板推送成功')

})

}

this.renderSuccessBoxShow = true

this.renderInventoryList = []

this.productRenderParams = { removeParams: [], aloneReplaceParams: [], groupReplaceParams: [], TextureReplaceParams: [] };

((objArr) => {

this.utils.myForEach(objArr, (valueOne) => {

this.utils.myForEach(this[valueOne], (valueTwo, index) => {

this.utils.myForEach(valueTwo, (val, count) => {

if (val !== -1) {

this.$set(this[valueOne][index], count, -1)

}

})

})

})

})(['removeIconArr', 'replaceIconArr'])

},

showPromptDialogBox(text) {

this.promptDialogText = text

this.promptDialogBoxShow = true

let self = this

setTimeout(() => { self.promptDialogBoxShow = false }, 1500)

},

watchMyTask() {

this.routerMini(this.isRender == -1 && this.isTask == 0?'/pages/my-tasks/my-tasks': '/pages/mine/my-favorite-fitment/my-favorite-fitment', 'tasks')

this.renderSuccessBoxShow = false

this.closeAllTypeList()

},

goToRecharge() {

this.routerMini('/pages/buy-points/buy-points', 'pay')

},

routerMini(path, type) {

if (this.isMiniprogram) {

wx.miniProgram.navigateTo({ url: path })

} else { // h5或者APP

let flag = storage.session.get('platformCode') === 'mobile2b' && storage.session.get('token') && this.isShowReplaceIcon

type === 'pay' ? (flag ? parent.postMessage('pay', this.mobileToBUrl) : null) : (type === 'tasks' ? (flag ? parent.postMessage('tasks', this.mobileToBUrl) : null) : null)

}

},

getuserIsPackageInMonthly() {

if (storage.session.get('platformCode') === 'mobile2b') {

this.API.getMobileToBPackageInMonthly()

.then((res) => {

this.isMember = ((['1', '2', '3', '4', '5'].findIndex((value) => {return value == res.obj.state}) !== -1) && res.success)

})

} else {

this.API.getuserIsPackageInMonthly()

.then((res) => {

this.isMember = (res.success && (res.obj.state === '2' || res.obj.state === '3'))

})

}

},

getMobileCode() {

if (!this.MobileNumberAllow || this.bindingPhoneGetCodeText !== '获取验证码') { return }

this.API.getMobileCode({

phoneNumber: this.bindingPhoneMobileNumber

})

.then((res) => {

if (res.success) {

let count = 61, $self = this

this.getCodeTimer = setInterval(() => {

count--

$self.bindingPhoneGetCodeText = count + 's'

if (count <= 0) {

clearInterval($self.getCodeTimer)

$self.bindingPhoneGetCodeText = '获取验证码'

}

}, 1000)

}

})

},

beginBindingMobile(type) {

if (this.bindingPhoneAllow) {

this.API.confirmBingingMolide({

mobile: this.bindingPhoneMobileNumber,

authCode: this.bindingPhoneMobileCode

})

.then((res) => {

if (res.success) {

this.showPromptDialogBox('绑定成功')

this.krpanoBindingMobileBoxShow = false

this.krpanoRenderBoxShow = true

} else {

this.showPromptDialogBox(res.message)

}

})

.catch(() => { this.showPromptDialogBox('绑定失败') })

}

},

newSearchProCategory() {

this.API.newSearchProCategory({

cid: '',

msgId: 'asdf',

type: 'A'

})

.then((res) => {

if (res.success) {

let proCategory = []

myForEach(res.datalist[0].ff, (value) => {

proCategory = proCategory.concat(value.ff)

})

this.proCategory = proCategory

} else {

this.proCategory = []

}

})

.catch(() => {

this.proCategory = []

})

},

pySort(l1, empty) {

var PinYin = {"a":"\u554a\u963f\u9515","ai":"\u57c3\u6328\u54ce\u5509\u54c0\u7691\u764c\u853c\u77ee\u827e\u788d\u7231\u9698\u8bf6\u6371\u55f3\u55cc\u5ad2\u7477\u66a7\u7839\u953f\u972d","an":"\u978d\u6c28\u5b89\u4ffa\u6309\u6697\u5cb8\u80fa\u6848\u8c19\u57ef\u63de\u72b4\u5eb5\u6849\u94f5\u9e4c\u9878\u9eef","ang":"\u80ae\u6602\u76ce","ao":"\u51f9\u6556\u71ac\u7ff1\u8884\u50b2\u5965\u61ca\u6fb3\u5773\u62d7\u55f7\u5662\u5c99\u5ed2\u9068\u5aaa\u9a9c\u8071\u87af\u93ca\u9ccc\u93d6","ba":"\u82ad\u634c\u6252\u53ed\u5427\u7b06\u516b\u75a4\u5df4\u62d4\u8dcb\u9776\u628a\u8019\u575d\u9738\u7f62\u7238\u8307\u83dd\u8406\u636d\u5c9c\u705e\u6777\u94af\u7c91\u9c85\u9b43","bai":"\u767d\u67cf\u767e\u6446\u4f70\u8d25\u62dc\u7a17\u859c\u63b0\u97b4","ban":"\u6591\u73ed\u642c\u6273\u822c\u9881\u677f\u7248\u626e\u62cc\u4f34\u74e3\u534a\u529e\u7eca\u962a\u5742\u8c73\u94a3\u7622\u764d\u8228","bang":"\u90a6\u5e2e\u6886\u699c\u8180\u7ed1\u68d2\u78c5\u868c\u9551\u508d\u8c24\u84a1\u8783","bao":"\u82de\u80de\u5305\u8912\u96f9\u4fdd\u5821\u9971\u5b9d\u62b1\u62a5\u66b4\u8c79\u9c8d\u7206\u52f9\u8446\u5b80\u5b62\u7172\u9e28\u8913\u8db5\u9f85","bo":"\u5265\u8584\u73bb\u83e0\u64ad\u62e8\u94b5\u6ce2\u535a\u52c3\u640f\u94c2\u7b94\u4f2f\u5e1b\u8236\u8116\u818a\u6e24\u6cca\u9a73\u4eb3\u8543\u5575\u997d\u6a97\u64d8\u7934\u94b9\u9e41\u7c38\u8ddb","bei":"\u676f\u7891\u60b2\u5351\u5317\u8f88\u80cc\u8d1d\u94a1\u500d\u72c8\u5907\u60eb\u7119\u88ab\u5b5b\u9642\u90b6\u57e4\u84d3\u5457\u602b\u6096\u789a\u9e4e\u8919\u943e","ben":"\u5954\u82ef\u672c\u7b28\u755a\u574c\u951b","beng":"\u5d29\u7ef7\u752d\u6cf5\u8e66\u8ff8\u552a\u5623\u750f","bi":"\u903c\u9f3b\u6bd4\u9119\u7b14\u5f7c\u78a7\u84d6\u853d\u6bd5\u6bd9\u6bd6\u5e01\u5e87\u75f9\u95ed\u655d\u5f0a\u5fc5\u8f9f\u58c1\u81c2\u907f\u965b\u5315\u4ef3\u4ffe\u8298\u835c\u8378\u5421\u54d4\u72f4\u5eb3\u610e\u6ed7\u6fde\u5f3c\u59a3\u5a62\u5b16\u74a7\u8d32\u7540\u94cb\u79d5\u88e8\u7b5a\u7b85\u7be6\u822d\u895e\u8df8\u9ac0","bian":"\u97ad\u8fb9\u7f16\u8d2c\u6241\u4fbf\u53d8\u535e\u8fa8\u8fa9\u8fab\u904d\u533e\u5f01\u82c4\u5fed\u6c74\u7f0f\u7178\u782d\u78a5\u7a39\u7a86\u8759\u7b3e\u9cca","biao":"\u6807\u5f6a\u8198\u8868\u5a4a\u9aa0\u98d1\u98d9\u98da\u706c\u9556\u9573\u762d\u88f1\u9cd4","bie":"\u9cd6\u618b\u522b\u762a\u8e69\u9cd8","bin":"\u5f6c\u658c\u6fd2\u6ee8\u5bbe\u6448\u50a7\u6d5c\u7f24\u73a2\u6ba1\u8191\u9554\u9acc\u9b13","bing":"\u5175\u51b0\u67c4\u4e19\u79c9\u997c\u70b3\u75c5\u5e76\u7980\u90b4\u6452\u7ee0\u678b\u69df\u71f9","bu":"\u6355\u535c\u54fa\u8865\u57e0\u4e0d\u5e03\u6b65\u7c3f\u90e8\u6016\u62ca\u535f\u900b\u74ff\u6661\u949a\u91ad","ca":"\u64e6\u5693\u7924","cai":"\u731c\u88c1\u6750\u624d\u8d22\u776c\u8e29\u91c7\u5f69\u83dc\u8521","can":"\u9910\u53c2\u8695\u6b8b\u60ed\u60e8\u707f\u9a96\u74a8\u7cb2\u9eea","cang":"\u82cd\u8231\u4ed3\u6ca7\u85cf\u4f27","cao":"\u64cd\u7cd9\u69fd\u66f9\u8349\u8279\u5608\u6f15\u87ac\u825a","ce":"\u5395\u7b56\u4fa7\u518c\u6d4b\u5202\u5e3b\u607b","ceng":"\u5c42\u8e6d\u564c","cha":"\u63d2\u53c9\u832c\u8336\u67e5\u78b4\u643d\u5bdf\u5c94\u5dee\u8be7\u7339\u9987\u6c4a\u59f9\u6748\u6942\u69ce\u6aab\u9497\u9538\u9572\u8869","chai":"\u62c6\u67f4\u8c7a\u4faa\u8308\u7625\u867f\u9f87","chan":"\u6400\u63ba\u8749\u998b\u8c17\u7f20\u94f2\u4ea7\u9610\u98a4\u5181\u8c04\u8c36\u8487\u5edb\u5fcf\u6f7a\u6fb6\u5b71\u7fbc\u5a75\u5b17\u9aa3\u89c7\u7985\u9561\u88e3\u87fe\u8e94","chang":"\u660c\u7316\u573a\u5c1d\u5e38\u957f\u507f\u80a0\u5382\u655e\u7545\u5531\u5021\u4f25\u9b2f\u82cc\u83d6\u5f9c\u6005\u60dd\u960a\u5a3c\u5ae6\u6636\u6c05\u9cb3","chao":"\u8d85\u6284\u949e\u671d\u5632\u6f6e\u5de2\u5435\u7092\u600a\u7ec9\u6641\u8016","che":"\u8f66\u626f\u64a4\u63a3\u5f7b\u6f88\u577c\u5c6e\u7817","chen":"\u90f4\u81e3\u8fb0\u5c18\u6668\u5ff1\u6c89\u9648\u8d81\u886c\u79f0\u8c0c\u62bb\u55d4\u5bb8\u741b\u6987\u809c\u80c2\u789c\u9f80","cheng":"\u6491\u57ce\u6a59\u6210\u5448\u4e58\u7a0b\u60e9\u6f84\u8bda\u627f\u901e\u9a8b\u79e4\u57d5\u5d4a\u5fb5\u6d48\u67a8\u67fd\u6a18\u665f\u584d\u77a0\u94d6\u88ce\u86cf\u9172","chi":"\u5403\u75f4\u6301\u5319\u6c60\u8fdf\u5f1b\u9a70\u803b\u9f7f\u4f88\u5c3a\u8d64\u7fc5\u65a5\u70bd\u50ba\u5880\u82aa\u830c\u640b\u53f1\u54e7\u557b\u55e4\u5f73\u996c\u6cb2\u5ab8\u6555\u80dd\u7719\u7735\u9e31\u761b\u892b\u86a9\u87ad\u7b1e\u7bea\u8c49\u8e05\u8e1f\u9b51","chong":"\u5145\u51b2\u866b\u5d07\u5ba0\u833a\u5fe1\u61a7\u94f3\u825f","chou":"\u62bd\u916c\u7574\u8e0c\u7a20\u6101\u7b79\u4ec7\u7ef8\u7785\u4e11\u4fe6\u5733\u5e31\u60c6\u6eb4\u59af\u7633\u96e0\u9c8b","chu":"\u81ed\u521d\u51fa\u6a71\u53a8\u8e87\u9504\u96cf\u6ec1\u9664\u695a\u7840\u50a8\u77d7\u6410\u89e6\u5904\u4e8d\u520d\u61b7\u7ecc\u6775\u696e\u6a17\u870d\u8e70\u9edc","chuan":"\u63e3\u5ddd\u7a7f\u693d\u4f20\u8239\u5598\u4e32\u63be\u821b\u60f4\u9044\u5ddb\u6c1a\u948f\u9569\u8221","chuang":"\u75ae\u7a97\u5e62\u5e8a\u95ef\u521b\u6006","chui":"\u5439\u708a\u6376\u9524\u5782\u9672\u68f0\u69cc","chun":"\u6625\u693f\u9187\u5507\u6df3\u7eaf\u8822\u4fc3\u83bc\u6c8c\u80ab\u6710\u9e51\u877d","chuo":"\u6233\u7ef0\u851f\u8fb6\u8f8d\u955e\u8e14\u9f8a","ci":"\u75b5\u8328\u78c1\u96cc\u8f9e\u6148\u74f7\u8bcd\u6b64\u523a\u8d50\u6b21\u8360\u5472\u5d6f\u9e5a\u8785\u7ccd\u8d91","cong":"\u806a\u8471\u56f1\u5306\u4ece\u4e1b\u506c\u82c1\u6dd9\u9aa2\u742e\u7481\u679e","cu":"\u51d1\u7c97\u918b\u7c07\u731d\u6b82\u8e59","cuan":"\u8e7f\u7be1\u7a9c\u6c46\u64ba\u6615\u7228","cui":"\u6467\u5d14\u50ac\u8106\u7601\u7cb9\u6dec\u7fe0\u8403\u60b4\u7480\u69b1\u96b9","cun":"\u6751\u5b58\u5bf8\u78cb\u5fd6\u76b4","cuo":"\u64ae\u6413\u63aa\u632b\u9519\u539d\u811e\u9509\u77ec\u75e4\u9e7e\u8e49\u8e9c","da":"\u642d\u8fbe\u7b54\u7629\u6253\u5927\u8037\u54d2\u55d2\u601b\u59b2\u75b8\u8921\u7b2a\u977c\u9791","dai":"\u5446\u6b79\u50a3\u6234\u5e26\u6b86\u4ee3\u8d37\u888b\u5f85\u902e\u6020\u57ed\u7519\u5454\u5cb1\u8fe8\u902f\u9a80\u7ed0\u73b3\u9edb","dan":"\u803d\u62c5\u4e39\u5355\u90f8\u63b8\u80c6\u65e6\u6c2e\u4f46\u60ee\u6de1\u8bde\u5f39\u86cb\u4ebb\u510b\u5369\u840f\u5556\u6fb9\u6a90\u6b9a\u8d55\u7708\u7605\u8043\u7baa","dang":"\u5f53\u6321\u515a\u8361\u6863\u8c20\u51fc\u83ea\u5b95\u7800\u94db\u88c6","dao":"\u5200\u6363\u8e48\u5012\u5c9b\u7977\u5bfc\u5230\u7a3b\u60bc\u9053\u76d7\u53e8\u5541\u5fc9\u6d2e\u6c18\u7118\u5fd1\u7e9b","de":"\u5fb7\u5f97\u7684\u951d","deng":"\u8e6c\u706f\u767b\u7b49\u77aa\u51f3\u9093\u5654\u5d9d\u6225\u78f4\u956b\u7c26","di":"\u5824\u4f4e\u6ef4\u8fea\u654c\u7b1b\u72c4\u6da4\u7fdf\u5ae1\u62b5\u5e95\u5730\u8482\u7b2c\u5e1d\u5f1f\u9012\u7f14\u6c10\u7c74\u8bcb\u8c1b\u90b8\u577b\u839c\u837b\u5600\u5a23\u67e2\u68e3\u89cc\u7825\u78b2\u7747\u955d\u7f9d\u9ab6","dian":"\u98a0\u6382\u6ec7\u7898\u70b9\u5178\u975b\u57ab\u7535\u4f43\u7538\u5e97\u60e6\u5960\u6dc0\u6bbf\u4e36\u963d\u576b\u57dd\u5dc5\u73b7\u765c\u766b\u7c1f\u8e2e","diao":"\u7889\u53fc\u96d5\u51cb\u5201\u6389\u540a\u9493\u8c03\u8f7a\u94de\u8729\u7c9c\u8c82","die":"\u8dcc\u7239\u789f\u8776\u8fed\u8c0d\u53e0\u4f5a\u57a4\u581e\u63f2\u558b\u6e2b\u8f76\u7252\u74de\u8936\u800b\u8e40\u9cbd\u9cce","ding":"\u4e01\u76ef\u53ee\u9489\u9876\u9f0e\u952d\u5b9a\u8ba2\u4e22\u4ec3\u5576\u738e\u815a\u7887\u753a\u94e4\u7594\u8035\u914a","dong":"\u4e1c\u51ac\u8463\u61c2\u52a8\u680b\u4f97\u606b\u51bb\u6d1e\u578c\u549a\u5cbd\u5cd2\u5902\u6c21\u80e8\u80f4\u7850\u9e2b","dou":"\u515c\u6296\u6597\u9661\u8c46\u9017\u75d8\u8538\u94ad\u7aa6\u7aac\u86aa\u7bfc\u9161","du":"\u90fd\u7763\u6bd2\u728a\u72ec\u8bfb\u5835\u7779\u8d4c\u675c\u9540\u809a\u5ea6\u6e21\u5992\u828f\u561f\u6e0e\u691f\u6a50\u724d\u8839\u7b03\u9ad1\u9ee9","duan":"\u7aef\u77ed\u953b\u6bb5\u65ad\u7f0e\u5f56\u6934\u7145\u7c16","dui":"\u5806\u5151\u961f\u5bf9\u603c\u619d\u7893","dun":"\u58a9\u5428\u8e72\u6566\u987f\u56e4\u949d\u76fe\u9041\u7096\u7818\u7905\u76f9\u9566\u8db8","duo":"\u6387\u54c6\u591a\u593a\u579b\u8eb2\u6735\u8dfa\u8235\u5241\u60f0\u5815\u5484\u54da\u7f0d\u67c1\u94ce\u88f0\u8e31","e":"\u86fe\u5ce8\u9e45\u4fc4\u989d\u8bb9\u5a25\u6076\u5384\u627c\u904f\u9102\u997f\u5669\u8c14\u57a9\u57ad\u82ca\u83aa\u843c\u5443\u6115\u5c59\u5a40\u8f6d\u66f7\u816d\u786a\u9507\u9537\u9e57\u989a\u9cc4","en":"\u6069\u84bd\u6441\u5514\u55ef","er":"\u800c\u513f\u8033\u5c14\u9975\u6d31\u4e8c\u8d30\u8fe9\u73e5\u94d2\u9e38\u9c95","fa":"\u53d1\u7f5a\u7b4f\u4f10\u4e4f\u9600\u6cd5\u73d0\u57a1\u781d","fan":"\u85e9\u5e06\u756a\u7ffb\u6a0a\u77fe\u9492\u7e41\u51e1\u70e6\u53cd\u8fd4\u8303\u8d29\u72af\u996d\u6cdb\u8629\u5e61\u72ad\u68b5\u6535\u71d4\u7548\u8e6f","fang":"\u574a\u82b3\u65b9\u80aa\u623f\u9632\u59a8\u4eff\u8bbf\u7eba\u653e\u531a\u90a1\u5f77\u94ab\u822b\u9c82","fei":"\u83f2\u975e\u5561\u98de\u80a5\u532a\u8bfd\u5420\u80ba\u5e9f\u6cb8\u8d39\u82be\u72d2\u60b1\u6ddd\u5983\u7ecb\u7eef\u69a7\u8153\u6590\u6249\u7953\u7829\u9544\u75f1\u871a\u7bda\u7fe1\u970f\u9cb1","fen":"\u82ac\u915a\u5429\u6c1b\u5206\u7eb7\u575f\u711a\u6c7e\u7c89\u594b\u4efd\u5fff\u6124\u7caa\u507e\u7035\u68fc\u610d\u9cbc\u9f22","feng":"\u4e30\u5c01\u67ab\u8702\u5cf0\u950b\u98ce\u75af\u70fd\u9022\u51af\u7f1d\u8bbd\u5949\u51e4\u4ff8\u9146\u8451\u6ca3\u781c","fu":"\u4f5b\u5426\u592b\u6577\u80a4\u5b75\u6276\u62c2\u8f90\u5e45\u6c1f\u7b26\u4f0f\u4fd8\u670d\u6d6e\u6daa\u798f\u88b1\u5f17\u752b\u629a\u8f85\u4fef\u91dc\u65a7\u812f\u8151\u5e9c\u8150\u8d74\u526f\u8986\u8d4b\u590d\u5085\u4ed8\u961c\u7236\u8179\u8d1f\u5bcc\u8ba3\u9644\u5987\u7f1a\u5490\u5310\u51eb\u90db\u8299\u82fb\u832f\u83a9\u83d4\u544b\u5e5e\u6ecf\u8274\u5b5a\u9a78\u7ec2\u6874\u8d59\u9efb\u9efc\u7f58\u7a03\u99a5\u864d\u86a8\u8709\u8760\u876e\u9eb8\u8dba\u8dd7\u9cc6","ga":"\u5676\u560e\u86e4\u5c2c\u5477\u5c15\u5c1c\u65ee\u9486","gai":"\u8be5\u6539\u6982\u9499\u76d6\u6e89\u4e10\u9654\u5793\u6224\u8d45\u80f2","gan":"\u5e72\u7518\u6746\u67d1\u7aff\u809d\u8d76\u611f\u79c6\u6562\u8d63\u5769\u82f7\u5c34\u64c0\u6cd4\u6de6\u6f89\u7ec0\u6a44\u65f0\u77f8\u75b3\u9150","gang":"\u5188\u521a\u94a2\u7f38\u809b\u7eb2\u5c97\u6e2f\u6206\u7f61\u9883\u7b7b","gong":"\u6760\u5de5\u653b\u529f\u606d\u9f9a\u4f9b\u8eac\u516c\u5bab\u5f13\u5de9\u6c5e\u62f1\u8d21\u5171\u857b\u5efe\u54a3\u73d9\u80b1\u86a3\u86e9\u89e5","gao":"\u7bd9\u768b\u9ad8\u818f\u7f94\u7cd5\u641e\u9550\u7a3f\u544a\u777e\u8bf0\u90dc\u84bf\u85c1\u7f1f\u69d4\u69c1\u6772\u9506","ge":"\u54e5\u6b4c\u6401\u6208\u9e3d\u80f3\u7599\u5272\u9769\u845b\u683c\u9601\u9694\u94ec\u4e2a\u5404\u9b32\u4ee1\u54ff\u5865\u55dd\u7ea5\u643f\u8188\u784c\u94ea\u9549\u88bc\u988c\u867c\u8238\u9abc\u9ac2","gei":"\u7ed9","gen":"\u6839\u8ddf\u4e98\u831b\u54cf\u826e","geng":"\u8015\u66f4\u5e9a\u7fb9\u57c2\u803f\u6897\u54fd\u8d53\u9ca0","gou":"\u94a9\u52fe\u6c9f\u82df\u72d7\u57a2\u6784\u8d2d\u591f\u4f5d\u8bdf\u5ca3\u9058\u5abe\u7f11\u89cf\u5f40\u9e32\u7b31\u7bdd\u97b2","gu":"\u8f9c\u83c7\u5495\u7b8d\u4f30\u6cbd\u5b64\u59d1\u9f13\u53e4\u86ca\u9aa8\u8c37\u80a1\u6545\u987e\u56fa\u96c7\u560f\u8bc2\u83f0\u54cc\u5d2e\u6c69\u688f\u8f71\u726f\u727f\u80cd\u81cc\u6bc2\u77bd\u7f5f\u94b4\u9522\u74e0\u9e2a\u9e44\u75fc\u86c4\u9164\u89da\u9cb4\u9ab0\u9e58","gua":"\u522e\u74dc\u5250\u5be1\u6302\u8902\u5366\u8bd6\u5471\u681d\u9e39","guai":"\u4e56\u62d0\u602a\u54d9","guan":"\u68fa\u5173\u5b98\u51a0\u89c2\u7ba1\u9986\u7f50\u60ef\u704c\u8d2f\u500c\u839e\u63bc\u6dab\u76e5\u9e73\u9ccf","guang":"\u5149\u5e7f\u901b\u72b7\u6844\u80f1\u7592","gui":"\u7470\u89c4\u572d\u7845\u5f52\u9f9f\u95fa\u8f68\u9b3c\u8be1\u7678\u6842\u67dc\u8dea\u8d35\u523d\u5326\u523f\u5e8b\u5b84\u59ab\u6867\u7085\u6677\u7688\u7c0b\u9c91\u9cdc","gun":"\u8f8a\u6eda\u68cd\u4e28\u886e\u7ef2\u78d9\u9ca7","guo":"\u9505\u90ed\u56fd\u679c\u88f9\u8fc7\u9998\u8803\u57da\u63b4\u5459\u56d7\u5e3c\u5d1e\u7313\u6901\u8662\u951e\u8052\u872e\u873e\u8748","ha":"\u54c8","hai":"\u9ab8\u5b69\u6d77\u6c26\u4ea5\u5bb3\u9a87\u54b4\u55e8\u988f\u91a2","han":"\u9163\u61a8\u90af\u97e9\u542b\u6db5\u5bd2\u51fd\u558a\u7f55\u7ff0\u64bc\u634d\u65f1\u61be\u608d\u710a\u6c57\u6c49\u9097\u83e1\u6496\u961a\u701a\u6657\u7113\u9894\u86b6\u9f3e","hen":"\u592f\u75d5\u5f88\u72e0\u6068","hang":"\u676d\u822a\u6c86\u7ed7\u73e9\u6841","hao":"\u58d5\u568e\u8c6a\u6beb\u90dd\u597d\u8017\u53f7\u6d69\u8585\u55e5\u5686\u6fe0\u704f\u660a\u7693\u98a2\u869d","he":"\u5475\u559d\u8377\u83cf\u6838\u79be\u548c\u4f55\u5408\u76d2\u8c89\u9602\u6cb3\u6db8\u8d6b\u8910\u9e64\u8d3a\u8bc3\u52be\u58d1\u85ff\u55d1\u55ec\u9616\u76cd\u86b5\u7fee","hei":"\u563f\u9ed1","heng":"\u54fc\u4ea8\u6a2a\u8861\u6052\u8a07\u8605","hong":"\u8f70\u54c4\u70d8\u8679\u9e3f\u6d2a\u5b8f\u5f18\u7ea2\u9ec9\u8ba7\u836d\u85a8\u95f3\u6cd3","hou":"\u5589\u4faf\u7334\u543c\u539a\u5019\u540e\u5820\u5f8c\u9005\u760a\u7bcc\u7cc7\u9c8e\u9aba","hu":"\u547c\u4e4e\u5ffd\u745a\u58f6\u846b\u80e1\u8774\u72d0\u7cca\u6e56\u5f27\u864e\u552c\u62a4\u4e92\u6caa\u6237\u51b1\u553f\u56eb\u5cb5\u7322\u6019\u60da\u6d52\u6ef9\u7425\u69f2\u8f77\u89f3\u70c0\u7173\u623d\u6248\u795c\u9e55\u9e71\u7b0f\u9190\u659b","hua":"\u82b1\u54d7\u534e\u733e\u6ed1\u753b\u5212\u5316\u8bdd\u5290\u6d4d\u9a85\u6866\u94e7\u7a1e","huai":"\u69d0\u5f8a\u6000\u6dee\u574f\u8fd8\u8e1d","huan":"\u6b22\u73af\u6853\u7f13\u6362\u60a3\u5524\u75ea\u8c62\u7115\u6da3\u5ba6\u5e7b\u90c7\u5942\u57b8\u64d0\u571c\u6d39\u6d63\u6f36\u5bf0\u902d\u7f33\u953e\u9ca9\u9b1f","huang":"\u8352\u614c\u9ec4\u78fa\u8757\u7c27\u7687\u51f0\u60f6\u714c\u6643\u5e4c\u604d\u8c0e\u968d\u5fa8\u6e5f\u6f62\u9051\u749c\u8093\u7640\u87e5\u7bc1\u9cc7","hui":"\u7070\u6325\u8f89\u5fbd\u6062\u86d4\u56de\u6bc1\u6094\u6167\u5349\u60e0\u6666\u8d3f\u79fd\u4f1a\u70e9\u6c47\u8bb3\u8bf2\u7ed8\u8bd9\u8334\u835f\u8559\u54d5\u5599\u96b3\u6d04\u5f57\u7f0b\u73f2\u6656\u605a\u867a\u87ea\u9ebe","hun":"\u8364\u660f\u5a5a\u9b42\u6d51\u6df7\u8be8\u9984\u960d\u6eb7\u7f17","huo":"\u8c41\u6d3b\u4f19\u706b\u83b7\u6216\u60d1\u970d\u8d27\u7978\u6509\u56af\u5925\u94ac\u952a\u956c\u8020\u8816","ji":"\u51fb\u573e\u57fa\u673a\u7578\u7a3d\u79ef\u7b95\u808c\u9965\u8ff9\u6fc0\u8ba5\u9e21\u59ec\u7ee9\u7f09\u5409\u6781\u68d8\u8f91\u7c4d\u96c6\u53ca\u6025\u75be\u6c72\u5373\u5ac9\u7ea7\u6324\u51e0\u810a\u5df1\u84df\u6280\u5180\u5b63\u4f0e\u796d\u5242\u60b8\u6d4e\u5bc4\u5bc2\u8ba1\u8bb0\u65e2\u5fcc\u9645\u5993\u7ee7\u7eaa\u5c45\u4e0c\u4e69\u525e\u4f76\u4f74\u8114\u58bc\u82a8\u82b0\u8401\u84ba\u857a\u638e\u53fd\u54ad\u54dc\u5527\u5c8c\u5d74\u6d0e\u5f50\u5c50\u9aa5\u757f\u7391\u696b\u6b9b\u621f\u6222\u8d4d\u89ca\u7284\u9f51\u77f6\u7f81\u5d47\u7a37\u7620\u7635\u866e\u7b08\u7b04\u66a8\u8dfb\u8dfd\u9701\u9c9a\u9cab\u9afb\u9e82","jia":"\u5609\u67b7\u5939\u4f73\u5bb6\u52a0\u835a\u988a\u8d3e\u7532\u94be\u5047\u7a3c\u4ef7\u67b6\u9a7e\u5ac1\u4f3d\u90cf\u62ee\u5cac\u6d43\u8fe6\u73c8\u621b\u80db\u605d\u94d7\u9553\u75c2\u86f1\u7b33\u8888\u8dcf","jian":"\u6b7c\u76d1\u575a\u5c16\u7b3a\u95f4\u714e\u517c\u80a9\u8270\u5978\u7f04\u8327\u68c0\u67ec\u78b1\u7877\u62e3\u6361\u7b80\u4fed\u526a\u51cf\u8350\u69db\u9274\u8df5\u8d31\u89c1\u952e\u7bad\u4ef6\u5065\u8230\u5251\u996f\u6e10\u6e85\u6da7\u5efa\u50ed\u8c0f\u8c2b\u83c5\u84b9\u641b\u56dd\u6e54\u8e47\u8b07\u7f23\u67a7\u67d9\u6957\u620b\u622c\u726e\u728d\u6bfd\u8171\u7751\u950f\u9e63\u88e5\u7b15\u7bb4\u7fe6\u8dbc\u8e3a\u9ca3\u97af","jiang":"\u50f5\u59dc\u5c06\u6d46\u6c5f\u7586\u848b\u6868\u5956\u8bb2\u5320\u9171\u964d\u8333\u6d1a\u7edb\u7f30\u729f\u7913\u8029\u7ce8\u8c47","jiao":"\u8549\u6912\u7901\u7126\u80f6\u4ea4\u90ca\u6d47\u9a84\u5a07\u56bc\u6405\u94f0\u77eb\u4fa5\u811a\u72e1\u89d2\u997a\u7f34\u7ede\u527f\u6559\u9175\u8f7f\u8f83\u53eb\u4f7c\u50ec\u832d\u6322\u564d\u5ce4\u5fbc\u59e3\u7e9f\u656b\u768e\u9e6a\u86df\u91ae\u8de4\u9c9b","jie":"\u7a96\u63ed\u63a5\u7686\u79f8\u8857\u9636\u622a\u52ab\u8282\u6854\u6770\u6377\u776b\u7aed\u6d01\u7ed3\u89e3\u59d0\u6212\u85c9\u82a5\u754c\u501f\u4ecb\u75a5\u8beb\u5c4a\u5048\u8ba6\u8bd8\u5588\u55df\u736c\u5a55\u5b51\u6840\u7352\u78a3\u9534\u7596\u88b7\u9889\u86a7\u7faf\u9c92\u9ab1\u9aeb","jin":"\u5dfe\u7b4b\u65a4\u91d1\u4eca\u6d25\u895f\u7d27\u9526\u4ec5\u8c28\u8fdb\u9773\u664b\u7981\u8fd1\u70ec\u6d78\u5c3d\u537a\u8369\u5807\u5664\u9991\u5ed1\u5997\u7f19\u747e\u69ff\u8d46\u89d0\u9485\u9513\u887f\u77dc","jing":"\u52b2\u8346\u5162\u830e\u775b\u6676\u9cb8\u4eac\u60ca\u7cbe\u7cb3\u7ecf\u4e95\u8b66\u666f\u9888\u9759\u5883\u656c\u955c\u5f84\u75c9\u9756\u7adf\u7ade\u51c0\u522d\u5106\u9631\u83c1\u734d\u61ac\u6cfe\u8ff3\u5f2a\u5a67\u80bc\u80eb\u8148\u65cc","jiong":"\u70af\u7a98\u5182\u8fe5\u6243","jiu":"\u63ea\u7a76\u7ea0\u7396\u97ed\u4e45\u7078\u4e5d\u9152\u53a9\u6551\u65e7\u81fc\u8205\u548e\u5c31\u759a\u50e6\u557e\u9604\u67e9\u6855\u9e6b\u8d73\u9b0f","ju":"\u97a0\u62d8\u72d9\u75bd\u9a79\u83ca\u5c40\u5480\u77e9\u4e3e\u6cae\u805a\u62d2\u636e\u5de8\u5177\u8ddd\u8e1e\u952f\u4ff1\u53e5\u60e7\u70ac\u5267\u5028\u8bb5\u82e3\u82f4\u8392\u63ac\u907d\u5c66\u741a\u67b8\u6910\u6998\u6989\u6a58\u728b\u98d3\u949c\u9514\u7aad\u88fe\u8d84\u91b5\u8e3d\u9f83\u96ce\u97ab","juan":"\u6350\u9e43\u5a1f\u5026\u7737\u5377\u7ee2\u9104\u72f7\u6d93\u684a\u8832\u9529\u954c\u96bd","jue":"\u6485\u652b\u6289\u6398\u5014\u7235\u89c9\u51b3\u8bc0\u7edd\u53a5\u5282\u8c32\u77cd\u8568\u5658\u5d1b\u7357\u5b53\u73cf\u6877\u6a5b\u721d\u9562\u8e76\u89d6","jun":"\u5747\u83cc\u94a7\u519b\u541b\u5cfb\u4fca\u7ae3\u6d5a\u90e1\u9a8f\u6343\u72fb\u76b2\u7b60\u9e87","ka":"\u5580\u5496\u5361\u4f67\u5494\u80e9","ke":"\u54af\u5777\u82db\u67ef\u68f5\u78d5\u9897\u79d1\u58f3\u54b3\u53ef\u6e34\u514b\u523b\u5ba2\u8bfe\u5ca2\u606a\u6e98\u9a92\u7f02\u73c2\u8f72\u6c2a\u778c\u94b6\u75b4\u7aa0\u874c\u9ac1","kai":"\u5f00\u63e9\u6977\u51ef\u6168\u5240\u57b2\u8488\u5ffe\u607a\u94e0\u950e","kan":"\u520a\u582a\u52d8\u574e\u780d\u770b\u4f83\u51f5\u83b0\u83b6\u6221\u9f9b\u77b0","kang":"\u5eb7\u6177\u7ce0\u625b\u6297\u4ea2\u7095\u5751\u4f09\u95f6\u94aa","kao":"\u8003\u62f7\u70e4\u9760\u5c3b\u6832\u7292\u94d0","ken":"\u80af\u5543\u57a6\u6073\u57a0\u88c9\u9880","keng":"\u542d\u5fd0\u94ff","kong":"\u7a7a\u6050\u5b54\u63a7\u5025\u5d06\u7b9c","kou":"\u62a0\u53e3\u6263\u5bc7\u82a4\u853b\u53e9\u770d\u7b58","ku":"\u67af\u54ed\u7a9f\u82e6\u9177\u5e93\u88e4\u5233\u5800\u55be\u7ed4\u9ab7","kua":"\u5938\u57ae\u630e\u8de8\u80ef\u4f89","kuai":"\u5757\u7b77\u4fa9\u5feb\u84af\u90d0\u8489\u72ef\u810d","kuan":"\u5bbd\u6b3e\u9acb","kuang":"\u5321\u7b50\u72c2\u6846\u77ff\u7736\u65f7\u51b5\u8bd3\u8bf3\u909d\u5739\u593c\u54d0\u7ea9\u8d36","kui":"\u4e8f\u76d4\u5cbf\u7aa5\u8475\u594e\u9b41\u5080\u9988\u6127\u6e83\u9997\u532e\u5914\u9697\u63c6\u55b9\u559f\u609d\u6126\u9615\u9035\u668c\u777d\u8069\u8770\u7bd1\u81fe\u8dec","kun":"\u5764\u6606\u6346\u56f0\u6083\u9603\u7428\u951f\u918c\u9cb2\u9ae1","kuo":"\u62ec\u6269\u5ed3\u9614\u86de","la":"\u5783\u62c9\u5587\u8721\u814a\u8fa3\u5566\u524c\u647a\u908b\u65ef\u782c\u760c","lai":"\u83b1\u6765\u8d56\u5d03\u5f95\u6d9e\u6fd1\u8d49\u7750\u94fc\u765e\u7c41","lan":"\u84dd\u5a6a\u680f\u62e6\u7bee\u9611\u5170\u6f9c\u8c30\u63fd\u89c8\u61d2\u7f06\u70c2\u6ee5\u5549\u5c9a\u61d4\u6f24\u6984\u6593\u7f71\u9567\u8934","lang":"\u7405\u6994\u72fc\u5eca\u90ce\u6717\u6d6a\u83a8\u8497\u5577\u9606\u9512\u7a02\u8782","lao":"\u635e\u52b3\u7262\u8001\u4f6c\u59e5\u916a\u70d9\u6d9d\u5520\u5d02\u6833\u94d1\u94f9\u75e8\u91aa","le":"\u52d2\u4e50\u808b\u4ec2\u53fb\u561e\u6cd0\u9cd3","lei":"\u96f7\u956d\u857e\u78ca\u7d2f\u5121\u5792\u64c2\u7c7b\u6cea\u7fb8\u8bd4\u837d\u54a7\u6f2f\u5ad8\u7f27\u6a91\u8012\u9179","ling":"\u68f1\u51b7\u62ce\u73b2\u83f1\u96f6\u9f84\u94c3\u4f36\u7f9a\u51cc\u7075\u9675\u5cad\u9886\u53e6\u4ee4\u9143\u5844\u82d3\u5464\u56f9\u6ce0\u7eeb\u67c3\u68c2\u74f4\u8046\u86c9\u7fce\u9cae","leng":"\u695e\u6123","li":"\u5398\u68a8\u7281\u9ece\u7bf1\u72f8\u79bb\u6f13\u7406\u674e\u91cc\u9ca4\u793c\u8389\u8354\u540f\u6817\u4e3d\u5389\u52b1\u783e\u5386\u5229\u5088\u4f8b\u4fd0\u75e2\u7acb\u7c92\u6ca5\u96b6\u529b\u7483\u54e9\u4fea\u4fda\u90e6\u575c\u82c8\u8385\u84e0\u85dc\u6369\u5456\u5533\u55b1\u7301\u6ea7\u6fa7\u9026\u5a0c\u5ae0\u9a8a\u7f21\u73de\u67a5\u680e\u8f79\u623e\u783a\u8a48\u7f79\u9502\u9e42\u75a0\u75ac\u86ce\u870a\u8821\u7b20\u7be5\u7c9d\u91b4\u8dde\u96f3\u9ca1\u9ce2\u9ee7","lian":"\u4fe9\u8054\u83b2\u8fde\u9570\u5ec9\u601c\u6d9f\u5e18\u655b\u8138\u94fe\u604b\u70bc\u7ec3\u631b\u8539\u5941\u6f4b\u6fc2\u5a08\u740f\u695d\u6b93\u81c1\u81a6\u88e2\u880a\u9ca2","liang":"\u7cae\u51c9\u6881\u7cb1\u826f\u4e24\u8f86\u91cf\u667e\u4eae\u8c05\u589a\u690b\u8e09\u9753\u9b49","liao":"\u64a9\u804a\u50da\u7597\u71ce\u5be5\u8fbd\u6f66\u4e86\u6482\u9563\u5ed6\u6599\u84fc\u5c25\u5639\u7360\u5bee\u7f2d\u948c\u9e69\u8022","lie":"\u5217\u88c2\u70c8\u52a3\u730e\u51bd\u57d2\u6d0c\u8d94\u8e90\u9b23","lin":"\u7433\u6797\u78f7\u9716\u4e34\u90bb\u9cde\u6dcb\u51db\u8d41\u541d\u853a\u5d99\u5eea\u9074\u6aa9\u8f9a\u77b5\u7cbc\u8e8f\u9e9f","liu":"\u6e9c\u7409\u69b4\u786b\u998f\u7559\u5218\u7624\u6d41\u67f3\u516d\u62a1\u507b\u848c\u6cd6\u6d4f\u905b\u9a9d\u7efa\u65d2\u7198\u950d\u954f\u9e68\u938f","long":"\u9f99\u804b\u5499\u7b3c\u7abf\u9686\u5784\u62e2\u9647\u5f04\u5785\u830f\u6cf7\u73d1\u680a\u80e7\u783b\u7643","lou":"\u697c\u5a04\u6402\u7bd3\u6f0f\u964b\u55bd\u5d5d\u9542\u7618\u8027\u877c\u9ac5","lu":"\u82a6\u5362\u9885\u5e90\u7089\u63b3\u5364\u864f\u9c81\u9e93\u788c\u9732\u8def\u8d42\u9e7f\u6f5e\u7984\u5f55\u9646\u622e\u5786\u6445\u64b8\u565c\u6cf8\u6e0c\u6f09\u7490\u680c\u6a79\u8f73\u8f82\u8f98\u6c07\u80ea\u9565\u9e2c\u9e6d\u7c0f\u823b\u9c88","lv":"\u9a74\u5415\u94dd\u4fa3\u65c5\u5c65\u5c61\u7f15\u8651\u6c2f\u5f8b\u7387\u6ee4\u7eff\u634b\u95fe\u6988\u8182\u7a06\u891b","luan":"\u5ce6\u5b6a\u6ee6\u5375\u4e71\u683e\u9e3e\u92ae","lue":"\u63a0\u7565\u950a","lun":"\u8f6e\u4f26\u4ed1\u6ca6\u7eb6\u8bba\u56f5","luo":"\u841d\u87ba\u7f57\u903b\u9523\u7ba9\u9aa1\u88f8\u843d\u6d1b\u9a86\u7edc\u502e\u8366\u645e\u7321\u6cfa\u6924\u8136\u9559\u7630\u96d2","ma":"\u5988\u9ebb\u739b\u7801\u8682\u9a6c\u9a82\u561b\u5417\u551b\u72b8\u5b37\u6769\u9ebd","mai":"\u57cb\u4e70\u9ea6\u5356\u8fc8\u8109\u52a2\u836c\u54aa\u973e","man":"\u7792\u9992\u86ee\u6ee1\u8513\u66fc\u6162\u6f2b\u8c29\u5881\u5e54\u7f26\u71b3\u9558\u989f\u87a8\u9cd7\u9794","mang":"\u8292\u832b\u76f2\u5fd9\u83bd\u9099\u6f2d\u6726\u786d\u87d2","meng":"\u6c13\u840c\u8499\u6aac\u76df\u9530\u731b\u68a6\u5b5f\u52d0\u750d\u77a2\u61f5\u791e\u867b\u8722\u8813\u824b\u8268\u9efe","miao":"\u732b\u82d7\u63cf\u7784\u85d0\u79d2\u6e3a\u5e99\u5999\u55b5\u9088\u7f08\u7f2a\u676a\u6dfc\u7707\u9e4b\u8731","mao":"\u8305\u951a\u6bdb\u77db\u94c6\u536f\u8302\u5192\u5e3d\u8c8c\u8d38\u4f94\u88a4\u52d6\u8306\u5cc1\u7441\u6634\u7266\u8004\u65c4\u61cb\u7780\u86d1\u8765\u87ca\u9ae6","me":"\u4e48","mei":"\u73ab\u679a\u6885\u9176\u9709\u7164\u6ca1\u7709\u5a92\u9541\u6bcf\u7f8e\u6627\u5bd0\u59b9\u5a9a\u5776\u8393\u5d4b\u7338\u6d7c\u6e44\u6963\u9545\u9e5b\u8882\u9b45","men":"\u95e8\u95f7\u4eec\u626a\u739f\u7116\u61d1\u9494","mi":"\u772f\u919a\u9761\u7cdc\u8ff7\u8c1c\u5f25\u7c73\u79d8\u89c5\u6ccc\u871c\u5bc6\u5e42\u8288\u5196\u8c27\u863c\u5627\u7315\u736f\u6c68\u5b93\u5f2d\u8112\u6549\u7cf8\u7e3b\u9e8b","mian":"\u68c9\u7720\u7ef5\u5195\u514d\u52c9\u5a29\u7f05\u9762\u6c94\u6e4e\u817c\u7704","mie":"\u8511\u706d\u54a9\u881b\u7bfe","min":"\u6c11\u62bf\u76bf\u654f\u60af\u95fd\u82e0\u5cb7\u95f5\u6cef\u73c9","ming":"\u660e\u879f\u9e23\u94ed\u540d\u547d\u51a5\u8317\u6e9f\u669d\u7791\u9169","miu":"\u8c2c","mo":"\u6478\u6479\u8611\u6a21\u819c\u78e8\u6469\u9b54\u62b9\u672b\u83ab\u58a8\u9ed8\u6cab\u6f20\u5bde\u964c\u8c1f\u8309\u84e6\u998d\u5aeb\u9546\u79e3\u763c\u8031\u87c6\u8c8a\u8c98","mou":"\u8c0b\u725f\u67d0\u53b6\u54de\u5a7a\u7738\u936a","mu":"\u62c7\u7261\u4ea9\u59c6\u6bcd\u5893\u66ae\u5e55\u52df\u6155\u6728\u76ee\u7766\u7267\u7a46\u4eeb\u82dc\u5452\u6c90\u6bea\u94bc","na":"\u62ff\u54ea\u5450\u94a0\u90a3\u5a1c\u7eb3\u5185\u637a\u80ad\u954e\u8872\u7bac","nai":"\u6c16\u4e43\u5976\u8010\u5948\u9f10\u827f\u8418\u67f0","nan":"\u5357\u7537\u96be\u56ca\u5583\u56e1\u6960\u8169\u877b\u8d67","nao":"\u6320\u8111\u607c\u95f9\u5b6c\u57b4\u7331\u7459\u7847\u94d9\u86f2","ne":"\u6dd6\u5462\u8bb7","nei":"\u9981","nen":"\u5ae9\u80fd\u6798\u6041","ni":"\u59ae\u9713\u502a\u6ce5\u5c3c\u62df\u4f60\u533f\u817b\u9006\u6eba\u4f32\u576d\u730a\u6029\u6ee0\u6635\u65ce\u7962\u615d\u7768\u94cc\u9cb5","nian":"\u852b\u62c8\u5e74\u78be\u64b5\u637b\u5ff5\u5eff\u8f87\u9ecf\u9c87\u9cb6","niang":"\u5a18\u917f","niao":"\u9e1f\u5c3f\u8311\u5b32\u8132\u8885","nie":"\u634f\u8042\u5b7d\u556e\u954a\u954d\u6d85\u4e5c\u9667\u8616\u55eb\u8080\u989e\u81ec\u8e51","nin":"\u60a8\u67e0","ning":"\u72de\u51dd\u5b81\u62e7\u6cde\u4f5e\u84e5\u549b\u752f\u804d","niu":"\u725b\u626d\u94ae\u7ebd\u72c3\u5ff8\u599e\u86b4","nong":"\u8113\u6d53\u519c\u4fac","nu":"\u5974\u52aa\u6012\u5476\u5e11\u5f29\u80ec\u5b65\u9a7d","nv":"\u5973\u6067\u9495\u8844","nuan":"\u6696","nuenue":"\u8650","nue":"\u759f\u8c11","nuo":"\u632a\u61e6\u7cef\u8bfa\u50a9\u6426\u558f\u9518","ou":"\u54e6\u6b27\u9e25\u6bb4\u85d5\u5455\u5076\u6ca4\u6004\u74ef\u8026","pa":"\u556a\u8db4\u722c\u5e15\u6015\u7436\u8469\u7b62","pai":"\u62cd\u6392\u724c\u5f98\u6e43\u6d3e\u4ff3\u848e","pan":"\u6500\u6f58\u76d8\u78d0\u76fc\u7554\u5224\u53db\u723f\u6cee\u88a2\u897b\u87e0\u8e52","pang":"\u4e53\u5e9e\u65c1\u802a\u80d6\u6ec2\u9004","pao":"\u629b\u5486\u5228\u70ae\u888d\u8dd1\u6ce1\u530f\u72cd\u5e96\u812c\u75b1","pei":"\u5478\u80da\u57f9\u88f4\u8d54\u966a\u914d\u4f69\u6c9b\u638a\u8f94\u5e14\u6de0\u65c6\u952b\u9185\u9708","pen":"\u55b7\u76c6\u6e53","peng":"\u7830\u62a8\u70f9\u6f8e\u5f6d\u84ec\u68da\u787c\u7bf7\u81a8\u670b\u9e4f\u6367\u78b0\u576f\u580b\u562d\u6026\u87db","pi":"\u7812\u9739\u6279\u62ab\u5288\u7435\u6bd7\u5564\u813e\u75b2\u76ae\u5339\u75de\u50fb\u5c41\u8b6c\u4e15\u9674\u90b3\u90eb\u572e\u9f19\u64d7\u567c\u5e80\u5ab2\u7eb0\u6787\u7513\u7765\u7f74\u94cd\u75e6\u7656\u758b\u868d\u8c94","pian":"\u7bc7\u504f\u7247\u9a97\u8c1d\u9a88\u728f\u80fc\u890a\u7fe9\u8e41","piao":"\u98d8\u6f02\u74e2\u7968\u527d\u560c\u5ad6\u7f25\u6b8d\u779f\u87b5","pie":"\u6487\u77a5\u4e3f\u82e4\u6c15","pin":"\u62fc\u9891\u8d2b\u54c1\u8058\u62da\u59d8\u5ad4\u6980\u725d\u98a6","ping":"\u4e52\u576a\u82f9\u840d\u5e73\u51ed\u74f6\u8bc4\u5c4f\u4fdc\u5a09\u67b0\u9c86","po":"\u5761\u6cfc\u9887\u5a46\u7834\u9b44\u8feb\u7c95\u53f5\u9131\u6ea5\u73c0\u948b\u94b7\u76a4\u7b38","pou":"\u5256\u88d2\u8e23","pu":"\u6251\u94fa\u4ec6\u8386\u8461\u83e9\u84b2\u57d4\u6734\u5703\u666e\u6d66\u8c31\u66dd\u7011\u530d\u5657\u6fee\u749e\u6c06\u9564\u9568\u8e7c","qi":"\u671f\u6b3a\u6816\u621a\u59bb\u4e03\u51c4\u6f06\u67d2\u6c8f\u5176\u68cb\u5947\u6b67\u7566\u5d0e\u8110\u9f50\u65d7\u7948\u7941\u9a91\u8d77\u5c82\u4e5e\u4f01\u542f\u5951\u780c\u5668\u6c14\u8fc4\u5f03\u6c7d\u6ce3\u8bab\u4e9f\u4e93\u573b\u8291\u840b\u847a\u5601\u5c7a\u5c90\u6c54\u6dc7\u9a90\u7eee\u742a\u7426\u675e\u6864\u69ed\u6b39\u797a\u61a9\u789b\u86f4\u871e\u7da6\u7dae\u8dbf\u8e4a\u9ccd\u9e92","qia":"\u6390\u6070\u6d3d\u845c","qian":"\u7275\u6266\u948e\u94c5\u5343\u8fc1\u7b7e\u4edf\u8c26\u4e7e\u9ed4\u94b1\u94b3\u524d\u6f5c\u9063\u6d45\u8c34\u5811\u5d4c\u6b20\u6b49\u4f65\u9621\u828a\u82a1\u8368\u63ae\u5c8d\u60ad\u614a\u9a9e\u6434\u8930\u7f31\u6920\u80b7\u6106\u94a4\u8654\u7b9d","qiang":"\u67aa\u545b\u8154\u7f8c\u5899\u8537\u5f3a\u62a2\u5af1\u6a2f\u6217\u709d\u9516\u9535\u956a\u8941\u8723\u7f9f\u8deb\u8dc4","qiao":"\u6a47\u9539\u6572\u6084\u6865\u77a7\u4e54\u4fa8\u5de7\u9798\u64ac\u7fd8\u5ced\u4fcf\u7a8d\u5281\u8bee\u8c2f\u835e\u6100\u6194\u7f32\u6a35\u6bf3\u7857\u8df7\u9792","qie":"\u5207\u8304\u4e14\u602f\u7a83\u90c4\u553c\u60ec\u59be\u6308\u9532\u7ba7","qin":"\u94a6\u4fb5\u4eb2\u79e6\u7434\u52e4\u82b9\u64d2\u79bd\u5bdd\u6c81\u82a9\u84c1\u8572\u63ff\u5423\u55ea\u5659\u6eb1\u6a8e\u8793\u887e","qing":"\u9752\u8f7b\u6c22\u503e\u537f\u6e05\u64ce\u6674\u6c30\u60c5\u9877\u8bf7\u5e86\u5029\u82d8\u570a\u6aa0\u78ec\u873b\u7f44\u7b90\u8b26\u9cad\u9ee5","qiong":"\u743c\u7a77\u909b\u8315\u7a79\u7b47\u928e","qiu":"\u79cb\u4e18\u90b1\u7403\u6c42\u56da\u914b\u6cc5\u4fc5\u6c3d\u5def\u827d\u72b0\u6e6b\u9011\u9052\u6978\u8d47\u9e20\u866c\u86af\u8764\u88d8\u7cd7\u9cc5\u9f3d","qu":"\u8d8b\u533a\u86c6\u66f2\u8eaf\u5c48\u9a71\u6e20\u53d6\u5a36\u9f8b\u8da3\u53bb\u8bce\u52ac\u8556\u8627\u5c96\u8862\u9612\u74a9\u89d1\u6c0d\u795b\u78f2\u766f\u86d0\u883c\u9eb4\u77bf\u9ee2","quan":"\u5708\u98a7\u6743\u919b\u6cc9\u5168\u75ca\u62f3\u72ac\u5238\u529d\u8be0\u8343\u737e\u609b\u7efb\u8f81\u754e\u94e8\u8737\u7b4c\u9b08","que":"\u7f3a\u7094\u7638\u5374\u9e4a\u69b7\u786e\u96c0\u9619\u60ab","qun":"\u88d9\u7fa4\u9021","ran":"\u7136\u71c3\u5189\u67d3\u82d2\u9aef","rang":"\u74e4\u58e4\u6518\u56b7\u8ba9\u79b3\u7a70","rao":"\u9976\u6270\u7ed5\u835b\u5a06\u6861","ruo":"\u60f9\u82e5\u5f31","re":"\u70ed\u504c","ren":"\u58ec\u4ec1\u4eba\u5fcd\u97e7\u4efb\u8ba4\u5203\u598a\u7eab\u4ede\u834f\u845a\u996a\u8f6b\u7a14\u887d","reng":"\u6254\u4ecd","ri":"\u65e5","rong":"\u620e\u8338\u84c9\u8363\u878d\u7194\u6eb6\u5bb9\u7ed2\u5197\u5d58\u72e8\u7f1b\u6995\u877e","rou":"\u63c9\u67d4\u8089\u7cc5\u8e42\u97a3","ru":"\u8339\u8815\u5112\u5b7a\u5982\u8fb1\u4e73\u6c5d\u5165\u8925\u84d0\u85b7\u5685\u6d33\u6ebd\u6fe1\u94f7\u8966\u98a5","ruan":"\u8f6f\u962e\u670a","rui":"\u854a\u745e\u9510\u82ae\u8564\u777f\u868b","run":"\u95f0\u6da6","sa":"\u6492\u6d12\u8428\u5345\u4ee8\u6332\u98d2","sai":"\u816e\u9cc3\u585e\u8d5b\u567b","san":"\u4e09\u53c1\u4f1e\u6563\u5f61\u9993\u6c35\u6bf5\u7cc1\u9730","sang":"\u6851\u55d3\u4e27\u6421\u78c9\u98a1","sao":"\u6414\u9a9a\u626b\u5ac2\u57fd\u81ca\u7619\u9ccb","se":"\u745f\u8272\u6da9\u556c\u94e9\u94ef\u7a51","sen":"\u68ee","seng":"\u50e7","sha":"\u838e\u7802\u6740\u5239\u6c99\u7eb1\u50bb\u5565\u715e\u810e\u6b43\u75e7\u88df\u970e\u9ca8","shai":"\u7b5b\u6652\u917e","shan":"\u73ca\u82eb\u6749\u5c71\u5220\u717d\u886b\u95ea\u9655\u64c5\u8d61\u81b3\u5584\u6c55\u6247\u7f2e\u5261\u8baa\u912f\u57cf\u829f\u6f78\u59d7\u9a9f\u81bb\u9490\u759d\u87ee\u8222\u8dda\u9cdd","shang":"\u5892\u4f24\u5546\u8d4f\u664c\u4e0a\u5c1a\u88f3\u57a7\u7ef1\u6b87\u71b5\u89de","shao":"\u68a2\u634e\u7a0d\u70e7\u828d\u52fa\u97f6\u5c11\u54e8\u90b5\u7ecd\u52ad\u82d5\u6f72\u86f8\u7b24\u7b72\u8244","she":"\u5962\u8d4a\u86c7\u820c\u820d\u8d66\u6444\u5c04\u6151\u6d89\u793e\u8bbe\u538d\u4f58\u731e\u7572\u9e9d","shen":"\u7837\u7533\u547b\u4f38\u8eab\u6df1\u5a20\u7ec5\u795e\u6c88\u5ba1\u5a76\u751a\u80be\u614e\u6e17\u8bdc\u8c02\u5432\u54c2\u6e16\u6939\u77e7\u8703","sheng":"\u58f0\u751f\u7525\u7272\u5347\u7ef3\u7701\u76db\u5269\u80dc\u5723\u4e1e\u6e11\u5ab5\u771a\u7b19","shi":"\u5e08\u5931\u72ee\u65bd\u6e7f\u8bd7\u5c38\u8671\u5341\u77f3\u62fe\u65f6\u4ec0\u98df\u8680\u5b9e\u8bc6\u53f2\u77e2\u4f7f\u5c4e\u9a76\u59cb\u5f0f\u793a\u58eb\u4e16\u67ff\u4e8b\u62ed\u8a93\u901d\u52bf\u662f\u55dc\u566c\u9002\u4ed5\u4f8d\u91ca\u9970\u6c0f\u5e02\u6043\u5ba4\u89c6\u8bd5\u8c25\u57d8\u83b3\u84cd\u5f11\u5511\u9963\u8f7c\u8006\u8d33\u70bb\u793b\u94c8\u94ca\u87ab\u8210\u7b6e\u8c55\u9ca5\u9cba","shou":"\u6536\u624b\u9996\u5b88\u5bff\u6388\u552e\u53d7\u7626\u517d\u624c\u72e9\u7ef6\u824f","shu":"\u852c\u67a2\u68b3\u6b8a\u6292\u8f93\u53d4\u8212\u6dd1\u758f\u4e66\u8d4e\u5b70\u719f\u85af\u6691\u66d9\u7f72\u8700\u9ecd\u9f20\u5c5e\u672f\u8ff0\u6811\u675f\u620d\u7ad6\u5885\u5eb6\u6570\u6f31\u6055\u500f\u587e\u83fd\u5fc4\u6cad\u6d91\u6f8d\u59dd\u7ebe\u6bf9\u8167\u6bb3\u956f\u79eb\u9e6c","shua":"\u5237\u800d\u5530\u6dae","shuai":"\u6454\u8870\u7529\u5e05\u87c0","shuan":"\u6813\u62f4\u95e9","shuang":"\u971c\u53cc\u723d\u5b40","shui":"\u8c01\u6c34\u7761\u7a0e","shun":"\u542e\u77ac\u987a\u821c\u6042","shuo":"\u8bf4\u7855\u6714\u70c1\u84b4\u6420\u55cd\u6fef\u5981\u69ca\u94c4","si":"\u65af\u6495\u5636\u601d\u79c1\u53f8\u4e1d\u6b7b\u8086\u5bfa\u55e3\u56db\u4f3a\u4f3c\u9972\u5df3\u53ae\u4fdf\u5155\u83e5\u549d\u6c5c\u6cd7\u6f8c\u59d2\u9a77\u7f0c\u7940\u7960\u9536\u9e36\u801c\u86f3\u7b25","song":"\u677e\u8038\u6002\u9882\u9001\u5b8b\u8bbc\u8bf5\u51c7\u83d8\u5d27\u5d69\u5fea\u609a\u6dde\u7ae6","sou":"\u641c\u8258\u64de\u55fd\u53df\u55d6\u55fe\u998a\u6eb2\u98d5\u778d\u953c\u878b","su":"\u82cf\u9165\u4fd7\u7d20\u901f\u7c9f\u50f3\u5851\u6eaf\u5bbf\u8bc9\u8083\u5919\u8c21\u850c\u55c9\u612b\u7c0c\u89eb\u7a23","suan":"\u9178\u849c\u7b97","sui":"\u867d\u968b\u968f\u7ee5\u9ad3\u788e\u5c81\u7a57\u9042\u96a7\u795f\u84d1\u51ab\u8c07\u6fc9\u9083\u71e7\u772d\u7762","sun":"\u5b59\u635f\u7b0b\u836a\u72f2\u98e7\u69ab\u8de3\u96bc","suo":"\u68ad\u5506\u7f29\u7410\u7d22\u9501\u6240\u5522\u55e6\u5a11\u686b\u7743\u7fa7","ta":"\u584c\u4ed6\u5b83\u5979\u5854\u736d\u631e\u8e4b\u8e0f\u95fc\u6ebb\u9062\u69bb\u6c93","tai":"\u80ce\u82d4\u62ac\u53f0\u6cf0\u915e\u592a\u6001\u6c70\u90b0\u85b9\u80bd\u70b1\u949b\u8dc6\u9c90","tan":"\u574d\u644a\u8d2a\u762b\u6ee9\u575b\u6a80\u75f0\u6f6d\u8c2d\u8c08\u5766\u6bef\u8892\u78b3\u63a2\u53f9\u70ad\u90ef\u8548\u6619\u94bd\u952c\u8983","tang":"\u6c64\u5858\u642a\u5802\u68e0\u819b\u5510\u7cd6\u50a5\u9967\u6e8f\u746d\u94f4\u9557\u8025\u8797\u87b3\u7fb0\u91a3","thang":"\u5018\u8eba\u6dcc","theng":"\u8d9f\u70eb","tao":"\u638f\u6d9b\u6ed4\u7ee6\u8404\u6843\u9003\u6dd8\u9676\u8ba8\u5957\u6311\u9f17\u5555\u97ec\u9955","te":"\u7279","teng":"\u85e4\u817e\u75bc\u8a8a\u6ed5","ti":"\u68af\u5254\u8e22\u9511\u63d0\u9898\u8e44\u557c\u4f53\u66ff\u568f\u60d5\u6d95\u5243\u5c49\u8351\u608c\u9016\u7ee8\u7f07\u9e48\u88fc\u918d","tian":"\u5929\u6dfb\u586b\u7530\u751c\u606c\u8214\u8146\u63ad\u5fdd\u9617\u6b84\u754b\u94bf\u86ba","tiao":"\u6761\u8fe2\u773a\u8df3\u4f7b\u7967\u94eb\u7a95\u9f86\u9ca6","tie":"\u8d34\u94c1\u5e16\u841c\u992e","ting":"\u5385\u542c\u70c3\u6c40\u5ef7\u505c\u4ead\u5ead\u633a\u8247\u839b\u8476\u5a77\u6883\u8713\u9706","tong":"\u901a\u6850\u916e\u77b3\u540c\u94dc\u5f64\u7ae5\u6876\u6345\u7b52\u7edf\u75db\u4f5f\u50ee\u4edd\u833c\u55f5\u6078\u6f7c\u783c","tou":"\u5077\u6295\u5934\u900f\u4ea0","tu":"\u51f8\u79c3\u7a81\u56fe\u5f92\u9014\u6d82\u5c60\u571f\u5410\u5154\u580d\u837c\u83df\u948d\u9174","tuan":"\u6e4d\u56e2\u7583","tui":"\u63a8\u9893\u817f\u8715\u892a\u9000\u5fd2\u717a","tun":"\u541e\u5c6f\u81c0\u9968\u66be\u8c5a\u7a80","tuo":"\u62d6\u6258\u8131\u9e35\u9640\u9a6e\u9a7c\u692d\u59a5\u62d3\u553e\u4e47\u4f57\u5768\u5eb9\u6cb1\u67dd\u7823\u7ba8\u8204\u8dce\u9f0d","wa":"\u6316\u54c7\u86d9\u6d3c\u5a03\u74e6\u889c\u4f64\u5a32\u817d","wai":"\u6b6a\u5916","wan":"\u8c4c\u5f2f\u6e7e\u73a9\u987d\u4e38\u70f7\u5b8c\u7897\u633d\u665a\u7696\u60cb\u5b9b\u5a49\u4e07\u8155\u525c\u8284\u82cb\u83c0\u7ea8\u7efe\u742c\u8118\u7579\u873f\u7ba2","wang":"\u6c6a\u738b\u4ea1\u6789\u7f51\u5f80\u65fa\u671b\u5fd8\u5984\u7f54\u5c22\u60d8\u8f8b\u9b4d","wei":"\u5a01\u5dcd\u5fae\u5371\u97e6\u8fdd\u6845\u56f4\u552f\u60df\u4e3a\u6f4d\u7ef4\u82c7\u840e\u59d4\u4f1f\u4f2a\u5c3e\u7eac\u672a\u851a\u5473\u754f\u80c3\u5582\u9b4f\u4f4d\u6e2d\u8c13\u5c09\u6170\u536b\u502d\u504e\u8bff\u9688\u8473\u8587\u5e0f\u5e37\u5d34\u5d6c\u7325\u732c\u95f1\u6ca9\u6d27\u6da0\u9036\u5a13\u73ae\u97ea\u8ece\u709c\u7168\u71a8\u75ff\u8249\u9c94","wen":"\u761f\u6e29\u868a\u6587\u95fb\u7eb9\u543b\u7a33\u7d0a\u95ee\u520e\u6120\u960c\u6c76\u74ba\u97eb\u6b81\u96ef","weng":"\u55e1\u7fc1\u74ee\u84ca\u8579","wo":"\u631d\u8717\u6da1\u7a9d\u6211\u65a1\u5367\u63e1\u6c83\u83b4\u5e44\u6e25\u674c\u809f\u9f8c","wu":"\u5deb\u545c\u94a8\u4e4c\u6c61\u8bec\u5c4b\u65e0\u829c\u68a7\u543e\u5434\u6bcb\u6b66\u4e94\u6342\u5348\u821e\u4f0d\u4fae\u575e\u620a\u96fe\u6664\u7269\u52ff\u52a1\u609f\u8bef\u5140\u4ef5\u9622\u90ac\u572c\u82b4\u5e91\u6003\u5fe4\u6d6f\u5be4\u8fd5\u59a9\u9a9b\u727e\u7110\u9e49\u9e5c\u8708\u92c8\u9f2f","xi":"\u6614\u7199\u6790\u897f\u7852\u77fd\u6670\u563b\u5438\u9521\u727a\u7a00\u606f\u5e0c\u6089\u819d\u5915\u60dc\u7184\u70ef\u6eaa\u6c50\u7280\u6a84\u88ad\u5e2d\u4e60\u5ab3\u559c\u94e3\u6d17\u7cfb\u9699\u620f\u7ec6\u50d6\u516e\u96b0\u90d7\u831c\u8478\u84f0\u595a\u550f\u5f99\u9969\u960b\u6d60\u6dc5\u5c63\u5b09\u73ba\u6a28\u66e6\u89cb\u6b37\u71b9\u798a\u79a7\u94b8\u7699\u7a78\u8725\u87cb\u823e\u7fb2\u7c9e\u7fd5\u91af\u9f37","xia":"\u778e\u867e\u5323\u971e\u8f96\u6687\u5ce1\u4fa0\u72ed\u4e0b\u53a6\u590f\u5413\u6380\u846d\u55c4\u72ce\u9050\u7455\u7856\u7615\u7f45\u9ee0","xian":"\u9528\u5148\u4ed9\u9c9c\u7ea4\u54b8\u8d24\u8854\u8237\u95f2\u6d8e\u5f26\u5acc\u663e\u9669\u73b0\u732e\u53bf\u817a\u9985\u7fa1\u5baa\u9677\u9650\u7ebf\u51bc\u85d3\u5c98\u7303\u66b9\u5a34\u6c19\u7946\u9e47\u75eb\u86ac\u7b45\u7c7c\u9170\u8df9","xiang":"\u76f8\u53a2\u9576\u9999\u7bb1\u8944\u6e58\u4e61\u7fd4\u7965\u8be6\u60f3\u54cd\u4eab\u9879\u5df7\u6a61\u50cf\u5411\u8c61\u8297\u8459\u9977\u5ea0\u9aa7\u7f03\u87d3\u9c9e\u98e8","xiao":"\u8427\u785d\u9704\u524a\u54ee\u56a3\u9500\u6d88\u5bb5\u6dc6\u6653\u5c0f\u5b5d\u6821\u8096\u5578\u7b11\u6548\u54d3\u54bb\u5d24\u6f47\u900d\u9a81\u7ee1\u67ad\u67b5\u7b71\u7bab\u9b48","xie":"\u6954\u4e9b\u6b47\u874e\u978b\u534f\u631f\u643a\u90aa\u659c\u80c1\u8c10\u5199\u68b0\u5378\u87f9\u61c8\u6cc4\u6cfb\u8c22\u5c51\u5055\u4eb5\u52f0\u71ee\u85a4\u64b7\u5ee8\u7023\u9082\u7ec1\u7f2c\u69ad\u698d\u6b59\u8e9e","xin":"\u85aa\u82af\u950c\u6b23\u8f9b\u65b0\u5ffb\u5fc3\u4fe1\u8845\u56df\u99a8\u8398\u6b46\u94fd\u946b","xing":"\u661f\u8165\u7329\u60fa\u5174\u5211\u578b\u5f62\u90a2\u884c\u9192\u5e78\u674f\u6027\u59d3\u9649\u8347\u8365\u64e4\u60bb\u784e","xiong":"\u5144\u51f6\u80f8\u5308\u6c79\u96c4\u718a\u828e","xiu":"\u4f11\u4fee\u7f9e\u673d\u55c5\u9508\u79c0\u8896\u7ee3\u83a0\u5cab\u9990\u5ea5\u9e3a\u8c85\u9af9","xu":"\u589f\u620c\u9700\u865a\u5618\u987b\u5f90\u8bb8\u84c4\u9157\u53d9\u65ed\u5e8f\u755c\u6064\u7d6e\u5a7f\u7eea\u7eed\u8bb4\u8be9\u5729\u84ff\u6035\u6d2b\u6e86\u987c\u6829\u7166\u7809\u76f1\u80e5\u7cc8\u9191","xuan":"\u8f69\u55a7\u5ba3\u60ac\u65cb\u7384\u9009\u7663\u7729\u7eda\u5107\u8c16\u8431\u63ce\u9994\u6ceb\u6d35\u6e32\u6f29\u7487\u6966\u6684\u70ab\u714a\u78b9\u94c9\u955f\u75c3","xue":"\u9774\u859b\u5b66\u7a74\u96ea\u8840\u5671\u6cf6\u9cd5","xun":"\u52cb\u718f\u5faa\u65ec\u8be2\u5bfb\u9a6f\u5de1\u6b89\u6c5b\u8bad\u8baf\u900a\u8fc5\u5dfd\u57d9\u8340\u85b0\u5ccb\u5f87\u6d54\u66db\u7aa8\u91ba\u9c9f","ya":"\u538b\u62bc\u9e26\u9e2d\u5440\u4e2b\u82bd\u7259\u869c\u5d16\u8859\u6daf\u96c5\u54d1\u4e9a\u8bb6\u4f22\u63e0\u5416\u5c88\u8fd3\u5a05\u740a\u6860\u6c29\u7811\u775a\u75d6","yan":"\u7109\u54bd\u9609\u70df\u6df9\u76d0\u4e25\u7814\u8712\u5ca9\u5ef6\u8a00\u989c\u960e\u708e\u6cbf\u5944\u63a9\u773c\u884d\u6f14\u8273\u5830\u71d5\u538c\u781a\u96c1\u5501\u5f66\u7130\u5bb4\u8c1a\u9a8c\u53a3\u9765\u8d5d\u4fe8\u5043\u5156\u8ba0\u8c33\u90fe\u9122\u82ab\u83f8\u5d26\u6079\u95eb\u960f\u6d07\u6e6e\u6edf\u598d\u5ae3\u7430\u664f\u80ed\u814c\u7131\u7f68\u7b75\u917d\u9b47\u990d\u9f39","yang":"\u6b83\u592e\u9e2f\u79e7\u6768\u626c\u4f6f\u75a1\u7f8a\u6d0b\u9633\u6c27\u4ef0\u75d2\u517b\u6837\u6f3e\u5f89\u600f\u6cf1\u7080\u70ca\u6059\u86d8\u9785","yao":"\u9080\u8170\u5996\u7476\u6447\u5c27\u9065\u7a91\u8c23\u59da\u54ac\u8200\u836f\u8981\u8000\u592d\u723b\u5406\u5d3e\u5fad\u7039\u5e7a\u73e7\u6773\u66dc\u80b4\u9e5e\u7a88\u7e47\u9cd0","ye":"\u6930\u564e\u8036\u7237\u91ce\u51b6\u4e5f\u9875\u6396\u4e1a\u53f6\u66f3\u814b\u591c\u6db2\u8c12\u90ba\u63f6\u9980\u6654\u70e8\u94d8","yi":"\u4e00\u58f9\u533b\u63d6\u94f1\u4f9d\u4f0a\u8863\u9890\u5937\u9057\u79fb\u4eea\u80f0\u7591\u6c82\u5b9c\u59e8\u5f5d\u6905\u8681\u501a\u5df2\u4e59\u77e3\u4ee5\u827a\u6291\u6613\u9091\u5c79\u4ebf\u5f79\u81c6\u9038\u8084\u75ab\u4ea6\u88d4\u610f\u6bc5\u5fc6\u4e49\u76ca\u6ea2\u8be3\u8bae\u8c0a\u8bd1\u5f02\u7ffc\u7fcc\u7ece\u5208\u5293\u4f7e\u8bd2\u572a\u572f\u57f8\u61ff\u82e1\u858f\u5f08\u5955\u6339\u5f0b\u5453\u54a6\u54bf\u566b\u5cc4\u5db7\u7317\u9974\u603f\u6021\u6092\u6f2a\u8fe4\u9a7f\u7f22\u6baa\u8d3b\u65d6\u71a0\u9487\u9552\u9571\u75cd\u7617\u7654\u7fca\u8864\u8734\u8223\u7fbf\u7ff3\u914f\u9edf","yin":"\u8335\u836b\u56e0\u6bb7\u97f3\u9634\u59fb\u541f\u94f6\u6deb\u5bc5\u996e\u5c39\u5f15\u9690\u5370\u80e4\u911e\u5819\u831a\u5591\u72fa\u5924\u6c24\u94df\u763e\u8693\u972a\u9f88","ying":"\u82f1\u6a31\u5a74\u9e70\u5e94\u7f28\u83b9\u8424\u8425\u8367\u8747\u8fce\u8d62\u76c8\u5f71\u9896\u786c\u6620\u5b34\u90e2\u8314\u83ba\u8426\u6484\u5624\u81ba\u6ee2\u6f46\u701b\u745b\u748e\u6979\u9e66\u763f\u988d\u7f42","yo":"\u54df\u5537","yong":"\u62e5\u4f63\u81c3\u75c8\u5eb8\u96cd\u8e0a\u86f9\u548f\u6cf3\u6d8c\u6c38\u607f\u52c7\u7528\u4fd1\u58c5\u5889\u6175\u9095\u955b\u752c\u9cd9\u9954","you":"\u5e7d\u4f18\u60a0\u5fe7\u5c24\u7531\u90ae\u94c0\u72b9\u6cb9\u6e38\u9149\u6709\u53cb\u53f3\u4f51\u91c9\u8bf1\u53c8\u5e7c\u5363\u6538\u4f91\u83b8\u5466\u56ff\u5ba5\u67da\u7337\u7256\u94d5\u75a3\u8763\u9c7f\u9edd\u9f2c","yu":"\u8fc2\u6de4\u4e8e\u76c2\u6986\u865e\u611a\u8206\u4f59\u4fde\u903e\u9c7c\u6109\u6e1d\u6e14\u9685\u4e88\u5a31\u96e8\u4e0e\u5c7f\u79b9\u5b87\u8bed\u7fbd\u7389\u57df\u828b\u90c1\u5401\u9047\u55bb\u5cea\u5fa1\u6108\u6b32\u72f1\u80b2\u8a89\u6d74\u5bd3\u88d5\u9884\u8c6b\u9a6d\u79ba\u6bd3\u4f1b\u4fe3\u8c00\u8c15\u8438\u84e3\u63c4\u5581\u5704\u5709\u5d5b\u72f3\u996b\u5ebe\u9608\u59aa\u59a4\u7ea1\u745c\u6631\u89ce\u8174\u6b24\u65bc\u715c\u71e0\u807f\u94b0\u9e46\u7610\u7600\u7ab3\u8753\u7afd\u8201\u96e9\u9f89","yuan":"\u9e33\u6e0a\u51a4\u5143\u57a3\u8881\u539f\u63f4\u8f95\u56ed\u5458\u5706\u733f\u6e90\u7f18\u8fdc\u82d1\u613f\u6028\u9662\u586c\u6c85\u5a9b\u7457\u6a7c\u7230\u7722\u9e22\u8788\u9f0b","yue":"\u66f0\u7ea6\u8d8a\u8dc3\u94a5\u5cb3\u7ca4\u6708\u60a6\u9605\u9fa0\u6a3e\u5216\u94ba","yun":"\u8018\u4e91\u90e7\u5300\u9668\u5141\u8fd0\u8574\u915d\u6655\u97f5\u5b55\u90d3\u82b8\u72c1\u607d\u7ead\u6b92\u6600\u6c32","za":"\u531d\u7838\u6742\u62f6\u5482","zai":"\u683d\u54c9\u707e\u5bb0\u8f7d\u518d\u5728\u54b1\u5d3d\u753e","zan":"\u6512\u6682\u8d5e\u74d2\u661d\u7c2a\u7ccc\u8db1\u933e","zang":"\u8d43\u810f\u846c\u5958\u6215\u81e7","zao":"\u906d\u7cdf\u51ff\u85fb\u67a3\u65e9\u6fa1\u86a4\u8e81\u566a\u9020\u7682\u7076\u71e5\u5523\u7f2b","ze":"\u8d23\u62e9\u5219\u6cfd\u4ec4\u8d5c\u5567\u8fee\u6603\u7b2e\u7ba6\u8234","zei":"\u8d3c","zen":"\u600e\u8c2e","zeng":"\u589e\u618e\u66fe\u8d60\u7f2f\u7511\u7f7e\u9503","zha":"\u624e\u55b3\u6e23\u672d\u8f67\u94e1\u95f8\u7728\u6805\u69a8\u548b\u4e4d\u70b8\u8bc8\u63f8\u5412\u54a4\u54f3\u600d\u781f\u75c4\u86b1\u9f44","zhai":"\u6458\u658b\u5b85\u7a84\u503a\u5be8\u7826","zhan":"\u77bb\u6be1\u8a79\u7c98\u6cbe\u76cf\u65a9\u8f97\u5d2d\u5c55\u8638\u6808\u5360\u6218\u7ad9\u6e5b\u7efd\u8c35\u640c\u65c3","zhang":"\u6a1f\u7ae0\u5f70\u6f33\u5f20\u638c\u6da8\u6756\u4e08\u5e10\u8d26\u4ed7\u80c0\u7634\u969c\u4ec9\u9123\u5e5b\u5d82\u7350\u5adc\u748b\u87d1","zhao":"\u62db\u662d\u627e\u6cbc\u8d75\u7167\u7f69\u5146\u8087\u53ec\u722a\u8bcf\u68f9\u948a\u7b0a","zhe":"\u906e\u6298\u54f2\u86f0\u8f99\u8005\u9517\u8517\u8fd9\u6d59\u8c2a\u966c\u67d8\u8f84\u78d4\u9e67\u891a\u8707\u8d6d","zhen":"\u73cd\u659f\u771f\u7504\u7827\u81fb\u8d1e\u9488\u4fa6\u6795\u75b9\u8bca\u9707\u632f\u9547\u9635\u7f1c\u6862\u699b\u8f78\u8d48\u80d7\u6715\u796f\u755b\u9e29","zheng":"\u84b8\u6323\u7741\u5f81\u72f0\u4e89\u6014\u6574\u62ef\u6b63\u653f\u5e27\u75c7\u90d1\u8bc1\u8be4\u5ce5\u94b2\u94ee\u7b5d","zhi":"\u829d\u679d\u652f\u5431\u8718\u77e5\u80a2\u8102\u6c41\u4e4b\u7ec7\u804c\u76f4\u690d\u6b96\u6267\u503c\u4f84\u5740\u6307\u6b62\u8dbe\u53ea\u65e8\u7eb8\u5fd7\u631a\u63b7\u81f3\u81f4\u7f6e\u5e1c\u5cd9\u5236\u667a\u79e9\u7a1a\u8d28\u7099\u75d4\u6ede\u6cbb\u7a92\u536e\u965f\u90c5\u57f4\u82b7\u646d\u5e19\u5fee\u5f58\u54ab\u9a98\u6809\u67b3\u6800\u684e\u8f75\u8f7e\u6534\u8d3d\u81a3\u7949\u7957\u9ef9\u96c9\u9e37\u75e3\u86ed\u7d77\u916f\u8dd6\u8e2c\u8e2f\u8c78\u89ef","zhong":"\u4e2d\u76c5\u5fe0\u949f\u8877\u7ec8\u79cd\u80bf\u91cd\u4ef2\u4f17\u51a2\u953a\u87bd\u8202\u822f\u8e35","zhou":"\u821f\u5468\u5dde\u6d32\u8bcc\u7ca5\u8f74\u8098\u5e1a\u5492\u76b1\u5b99\u663c\u9aa4\u5544\u7740\u501c\u8bf9\u836e\u9b3b\u7ea3\u80c4\u78a1\u7c40\u8233\u914e\u9cb7","zhu":"\u73e0\u682a\u86db\u6731\u732a\u8bf8\u8bdb\u9010\u7af9\u70db\u716e\u62c4\u77a9\u5631\u4e3b\u8457\u67f1\u52a9\u86c0\u8d2e\u94f8\u7b51\u4f4f\u6ce8\u795d\u9a7b\u4f2b\u4f8f\u90be\u82ce\u8331\u6d19\u6e1a\u6f74\u9a7a\u677c\u69e0\u6a65\u70b7\u94e2\u75b0\u7603\u86b0\u7afa\u7bb8\u7fe5\u8e85\u9e88","zhua":"\u6293","zhuai":"\u62fd","zhuan":"\u4e13\u7816\u8f6c\u64b0\u8d5a\u7bc6\u629f\u556d\u989b","zhuang":"\u6869\u5e84\u88c5\u5986\u649e\u58ee\u72b6\u4e2c","zhui":"\u690e\u9525\u8ffd\u8d58\u5760\u7f00\u8411\u9a93\u7f12","zhun":"\u8c06\u51c6","zhuo":"\u6349\u62d9\u5353\u684c\u7422\u8301\u914c\u707c\u6d4a\u502c\u8bfc\u5ef4\u855e\u64e2\u555c\u6d5e\u6dbf\u6753\u712f\u799a\u65ab","zi":"\u5179\u54a8\u8d44\u59ff\u6ecb\u6dc4\u5b5c\u7d2b\u4ed4\u7c7d\u6ed3\u5b50\u81ea\u6e0d\u5b57\u8c18\u5d6b\u59ca\u5b73\u7f01\u6893\u8f8e\u8d40\u6063\u7726\u9531\u79ed\u8014\u7b2b\u7ca2\u89dc\u8a3e\u9cbb\u9aed","zong":"\u9b03\u68d5\u8e2a\u5b97\u7efc\u603b\u7eb5\u8159\u7cbd","zou":"\u90b9\u8d70\u594f\u63cd\u9139\u9cb0","zu":"\u79df\u8db3\u5352\u65cf\u7956\u8bc5\u963b\u7ec4\u4fce\u83f9\u5550\u5f82\u9a75\u8e74","zuan":"\u94bb\u7e82\u6525\u7f35","zui":"\u5634\u9189\u6700\u7f6a","zun":"\u5c0a\u9075\u6499\u6a3d\u9cdf","zuo":"\u6628\u5de6\u4f50\u67de\u505a\u4f5c\u5750\u5ea7\u961d\u963c\u80d9\u795a\u9162","cou":"\u85ae\u6971\u8f8f\u8160","nang":"\u652e\u54dd\u56d4\u9995\u66e9","o":"\u5594","dia":"\u55f2","chuai":"\u562c\u81aa\u8e39","cen":"\u5c91\u6d94","diu":"\u94e5","nou":"\u8028","fou":"\u7f36","bia":"\u9adf"};

var l2 = l1.length;

var I1 = "";

var reg = new RegExp('[a-zA-Z0-9\- ]');

for (var i=0; i<l2; i++) {

var val = l1.substr(i,1);

var name = arraySearch(val,PinYin);

if(reg.test(val)) {

I1 += val;

} else if (name!==false) {

I1 += name;

}

}

I1 = I1.replace(/ /g,'-')

while (I1.indexOf('--')>0) {

I1 = I1.replace('--','-')

}

return I1;

function arraySearch(l1,l2){

for (var name in PinYin){

if (PinYin[name].indexOf(l1)!=-1) {

return ucfirst(name); break;

}

}

return false;

}

function ucfirst(l1){

if (l1.length>0) {

var first = l1.substr(0,1).toUpperCase();

var spare = l1.substr(1,l1.length);

return first + spare;

}

}

},

isChinese(temp) {

var re=/[^\u4E00-\u9FA5]/;

if (re.test(temp)){return false;}

return true ;

},

isChar(char) {

var reg = /[A-Za-z]/;

if (!reg.test(char)){return false ;}

return true ;

},

setKrpanoRenderListShow(type) {

let arr = []

this.utils.myForEach(this.replaceIconArr, (value, index) => {

this.utils.myForEach(value, (val, indexC) => {

if (val !== -1) {

let aloneObj = this.filterGoodsList(this.productsCostTypeList[index].detailList)[indexC]

aloneObj.indexP = index

aloneObj.indexC = indexC

aloneObj.type = 'replace'

arr.push(aloneObj)

} else if (this.removeIconArr[index][indexC] !== -1){

let aloneObj = this.filterGoodsList(this.productsCostTypeList[index].detailList)[indexC]

aloneObj.indexP = index

aloneObj.indexC = indexC

aloneObj.type = 'remove'

arr.push(aloneObj)

}

})

})

this.renderInventoryList = arr

type === 'recover' ? '' : this.AsRequiredClose('krpanoRenderListShow')

},

watchMobilePostMessage() {

window.addEventListener('message', (e) => {

if (e.data == 'paySuccess') {

this.isShowPayAmountStylePopBox = false

this.isShowPaySuccessPopBox = true

} else if (e.data == 'payFail'){

this.isShowPayAmountStylePopBox = false

this.isShowPayFailPopBox = true

}

})

},

isMiniprogramFn() {

const $self = this

function ready() { $self.isMiniprogram = (window.\_\_wxjs\_environment === 'miniprogram'), console.log('完成1', $self.isMiniprogram)}

(!window.WeixinJSBridge || !WeixinJSBridge.invoke)? document.addEventListener('WeixinJSBridgeReady', ready, false) : ready()

},

phoneOrPcDispose() {

const $self = this

if (document.documentElement.clientWidth >= 768) {

this.$set(this.swiperOption, 'allowTouchMove', false)

} else {

this.isiphoneOrPC = true

this.$nextTick(() => {

document.getElementById('pano').addEventListener('touchstart', () => {

$self.krpanoNavigationShow = true

$self.houseSwiperShow = false

})

})

}

this.$nextTick(() => {

document.querySelector('.houseSwitch-swiper .swiper-wrapper').style.margin = '0 auto'

})

this.goodsDetailsImgSwiperOption.on.slideChangeTransitionEnd = function() {

$self.goodsDetailsImgActiveIndex = this.activeIndex + 1

}

},

renovationMyHouse() {

if (this.isMiniprogram) {

switch (this.isRender) {

case '0' : wx.miniProgram.navigateTo({ url: '/pages/search-houseType/search-houseType?type=plan&source=webH5' }); break;

case '1' : this.putInMyHouse(); break;

case '2' : this.forbidReplaceShow = true; this.promptBoxText = '该空间正在装修中，请选择其他空间装修。'; break;

}

} else {

}

},

chooseRetainOrUnretain(type) {

this.fullHousePlanAction = type === 'left' ? 2 : 3

this.retainOriginalCaseBoxShow = false

this.confirmBeginRender()

},

goOnPutInHouse() {

this.fitmentFinishBoxShow = false

this.isMiniprogram ? wx.miniProgram.navigateBack() : null

},

watchRenderResult() {

this.fitmentFinishBoxShow = false

this.isMiniprogram ? this.routerMini('/pages/mine/my-favorite-fitment/my-favorite-fitment') : null

}

}

}