using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using System;

public class UIManager

{

static UIManager instance;

Dictionary<string, string> PrefabNameDic = new Dictionary<string, string>();//对应的预制体名

Dictionary<string, UIBaseWin> OpenWinsDic = new Dictionary<string, UIBaseWin>();//已生成的UI界面

//是否拦截鼠标键盘事件

public bool mInterceptEvent { get; set; }

public static UIManager Instance

{

get

{

if (instance == null)

instance = new UIManager();

return instance;

}

}

public UIManager()

{

PrefabNameDic.Add("Login","uiLoginWin");

PrefabNameDic.Add("Upload", "uiUploadWin");

PrefabNameDic.Add("UploadMat", "uiUploadMatWin");

PrefabNameDic.Add("UploadHard", "uiUploadHardWin");

PrefabNameDic.Add("UploadResult", "uiUploadResultWin");

PrefabNameDic.Add("GroupProduct", "uiGroupProductWin");

PrefabNameDic.Add("UploadGroup", "uiUploadGroupWin");

PrefabNameDic.Add("Progress", "uiProgressWin");

}

/// <summary>

/// 显示UI界面，如果是第一次打开生成一个

/// </summary>

/// <param name="key"></param>

public void Open(string key)

{

if (!PrefabNameDic.ContainsKey(key))

return;

if (OpenWinsDic.ContainsKey(key))

{

OpenWinsDic[key].Open();

}

else

{

Transform parent = GameObject.Find("Canvas").transform;

GameObject go = GameObject.Instantiate(Resources.Load("windows/" + PrefabNameDic[key]), parent) as GameObject;

UIBaseWin win = go.GetComponent<UIBaseWin>();

OpenWinsDic.Add(key, win);

}

UIManager.Instance.mInterceptEvent = true;

}

public void Close(string key)

{

if (!PrefabNameDic.ContainsKey(key))

return;

if (OpenWinsDic.ContainsKey(key))

{

if(OpenWinsDic[key] != null)

OpenWinsDic[key].Close();

}

mInterceptEvent = false;

foreach(UIBaseWin ui in OpenWinsDic.Values)

{

if(ui.gameObject.activeInHierarchy)

{

mInterceptEvent = true;

break;

}

}

}

public UIBaseWin GetWin(string key)

{

if (!OpenWinsDic.ContainsKey(key))

return null;

return OpenWinsDic[key];

}

public void Clear(bool deleteAll = false)

{

OpenWinsDic.Clear();

mInterceptEvent = false;

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

/// <summary>

/// UI基类

/// </summary>

public class UIBaseWin : MonoBehaviour

{

//[HideInInspector]

//string winName = string.Empty;

private void Awake()

{

//DontDestroyOnLoad(gameObject);

}

public void Open()

{

gameObject.SetActive(true);

}

public void Close()

{

gameObject.SetActive(false);

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.Events;

using System;

public class UICategory : MonoBehaviour

{

GameObject mBlocker;

List<UICategoryView> mViewList = new List<UICategoryView>();

public Text mCategoryText;

string[] mCategoryArray= { };

[HideInInspector]

public string mCategoryCode = "";

public UICategoryView mViewTemplate;

bool mIsShow = false;

public GameObject mPlaceHolder;

// Use this for initialization

void Start ()

{

}

private void OnEnable()

{

for (int i = mViewList.Count - 1; i >= 0; i--)

{

if (mViewList[i] != null)

GameObject.Destroy(mViewList[i].gameObject);

}

mViewList.Clear();

SetCategory(PlayerPrefs.GetString("Category"));

}

public void Show()

{

if (!mIsShow)

{

mIsShow = true;

CreateView(0, NetConst.Instance.mCategoryVO);

if (mViewList.Count > 0)

CreateBlocker(mViewList[0].mItemTemplate.gameObject);

}

}

public void Hide()

{

for (int i = mViewList.Count - 1; i >= 0; i--)

GameObject.Destroy(mViewList[i].gameObject);

mViewList.Clear();

DestroyBlocker();

mIsShow = false;

}

protected virtual void CreateBlocker(GameObject go)

{

// Get root Canvas.

var list = MyListPool<Canvas>.Get();

go.GetComponentsInParent(false, list);

if (list.Count == 0)

return;

Canvas rootCanvas = list[list.Count-1];

Canvas categoryCanvas = list[0];

MyListPool<Canvas>.Release(list);

Transform root = rootCanvas.transform;

// Create blocker GameObject.

mBlocker = new GameObject("Blocker");

// Setup blocker RectTransform to cover entire root canvas area.

RectTransform blockerRect = mBlocker.AddComponent<RectTransform>();

blockerRect.SetParent(root, false);

blockerRect.anchorMin = Vector3.zero;

blockerRect.anchorMax = Vector3.one;

blockerRect.sizeDelta = Vector2.zero;

// Make blocker be in separate canvas in same layer as dropdown and in layer just below it.

Canvas blockerCanvas = mBlocker.AddComponent<Canvas>();

blockerCanvas.overrideSorting = true;

blockerCanvas.sortingLayerID = categoryCanvas.sortingLayerID;

blockerCanvas.sortingOrder = categoryCanvas.sortingOrder - 1;

// Add raycaster since it's needed to block.

mBlocker.AddComponent<GraphicRaycaster>();

// Add image since it's needed to block, but make it clear.

Image blockerImage = mBlocker.AddComponent<Image>();

blockerImage.color = Color.clear;

// Add button since it's needed to block, and to close the dropdown when blocking area is clicked.

Button blockerButton = mBlocker.AddComponent<Button>();

blockerButton.onClick.AddListener(Hide);

}

protected virtual void DestroyBlocker()

{

if (mBlocker != null)

mBlocker.GetComponent<Button>().onClick.RemoveAllListeners();

Destroy(mBlocker);

}

public void NextPage(UICategoryItem item)

{

if (item.mVO.children == null)

{

CalcCategoryText();

Hide();

return;

}

for (int i = mViewList.Count - 1; i >= 0; i--)

{

if (mViewList[i] == null)

{

mViewList.RemoveAt(i);

}

else if (mViewList[i].mLayer > item.mView.mLayer)

{

GameObject.Destroy(mViewList[i].gameObject);

mViewList.RemoveAt(i);

}

}

CreateView(item.mView.mLayer+1,item.mVO,false);

}

public void CreateView(int layer, UICategoryVO1 vo, bool createChild = true)

{

GameObject go = GameObject.Instantiate(mViewTemplate.gameObject, mViewTemplate.transform.parent);

go.SetActive(true);

UICategoryView uiCV = go.GetComponent<UICategoryView>();

uiCV.SetLayer(layer);

mViewList.Add(uiCV);

uiCV.SetVO(vo);

if (createChild && mCategoryArray.Length > uiCV.mLayer)

uiCV.SelectItemByName(mCategoryArray[uiCV.mLayer]);

float uiTemplateWidth = mViewTemplate.GetComponent<RectTransform>().rect.width;

go.transform.localPosition = mViewTemplate.transform.localPosition +

new Vector3(uiTemplateWidth \* layer, 0, 0);

Canvas Canvas = go.AddComponent<Canvas>();

Canvas.overrideSorting = true;

Canvas.sortingOrder = 30000;

go.AddComponent<GraphicRaycaster>();

go.AddComponent<CanvasGroup>();

//显示下一级下拉框

if (createChild)

{

if (uiCV.mSelectedItem != null && uiCV.mSelectedItem.mVO.children != null)

CreateView(uiCV.mLayer + 1, uiCV.mSelectedItem.mVO);

}

}

public void CalcCategoryText()

{

string category = mViewList[0].mSelectedItem.mText.text;

for (int i = 1; i < mViewList.Count; i++)

{

category += "\\" + mViewList[i].mSelectedItem.mText.text;

}

SetCategory(category);

}

public void SetCategory(string txt)

{

if (mPlaceHolder != null)

{

if (txt == "")

mPlaceHolder.SetActive(true);

else

mPlaceHolder.SetActive(false);

}

//Debug.Log("SetCategory=" + txt);

mCategoryText.text = txt;

mCategoryArray = mCategoryText.text.Split('\\');

mCategoryCode = GetCategoryCode(NetConst.Instance.mCategoryVO.children, mCategoryArray, 0);

if(mCategoryCode == "" || mCategoryCode.EndsWith(","))//和模型分类对应不上

{

Reset();

}

PlayerPrefs.SetString("Category", mCategoryText.text);

//Debug.Log("mCategoryCode=" + mCategoryCode);

}

public void Reset()

{

mCategoryText.text = "";

Array.Clear(mCategoryArray, 0, mCategoryArray.Length); ;

mCategoryCode = "";

PlayerPrefs.SetString("Category", "");

mPlaceHolder.SetActive(true);

}

string GetCategoryCode(List<UICategoryVO1> list, string[] names, int index)

{

foreach (UICategoryVO1 vo in list)

{

if (vo.name == names[index])

{

if (vo.children == null)

return vo.keyCode;

else

return vo.keyCode + "," + GetCategoryCode(vo.children, names, index + 1);

}

}

return "";

}

}

static class MyListPool<T>

{

// Object pool to avoid allocations.

private static readonly MyObjectPool<List<T>> s\_ListPool = new MyObjectPool<List<T>>(null, l => l.Clear());

public static List<T> Get()

{

return s\_ListPool.Get();

}

public static void Release(List<T> toRelease)

{

s\_ListPool.Release(toRelease);

}

}

class MyObjectPool<T> where T : new()

{

private readonly Stack<T> m\_Stack = new Stack<T>();

private readonly UnityAction<T> m\_ActionOnGet;

private readonly UnityAction<T> m\_ActionOnRelease;

public int countAll { get; private set; }

public int countActive { get { return countAll - countInactive; } }

public int countInactive { get { return m\_Stack.Count; } }

public MyObjectPool(UnityAction<T> actionOnGet, UnityAction<T> actionOnRelease)

{

m\_ActionOnGet = actionOnGet;

m\_ActionOnRelease = actionOnRelease;

}

public T Get()

{

T element;

if (m\_Stack.Count == 0)

{

element = new T();

countAll++;

}

else

{

element = m\_Stack.Pop();

}

if (m\_ActionOnGet != null)

m\_ActionOnGet(element);

return element;

}

public void Release(T element)

{

if (m\_Stack.Count > 0 && ReferenceEquals(m\_Stack.Peek(), element))

Debug.LogError("Internal error. Trying to destroy object that is already released to pool.");

if (m\_ActionOnRelease != null)

m\_ActionOnRelease(element);

m\_Stack.Push(element);

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.EventSystems;

public class UICategoryItem : MonoBehaviour, IScrollHandler

{

public UICategoryVO1 mVO { get; set; }

[HideInInspector]

public Toggle mToggle { get; set; }

public Image mArrow;

public Text mText;

public UICategoryView mView;

Color32 mOriginColor = new Color32(0, 0, 0, 255);

Color32 mHighLightColor = new Color32(255,100,25,255);

public bool mSelected { get; set; }

public int mIndex{get;set;}

// Use this for initialization

private void Awake()

{

mToggle = GetComponent<Toggle>();

}

void Start ()

{

}

public void SetVO(UICategoryVO1 vo)

{

mVO = vo;

mText.text = mVO.name;

gameObject.name = mVO.name;

if (mVO.children != null)

mArrow.enabled = true;

else

mArrow.enabled = false;

}

public void SetIndex(int index)

{

mIndex = index;

}

public void OnHighLight(bool light)

{

mSelected = light;

if (light)

mText.color = mHighLightColor;

else

mText.color = mOriginColor;

}

public void OnMouseOver()

{

mText.color = mHighLightColor;

}

public void OnMouseExit()

{

if (mSelected)

mText.color = mHighLightColor;

else

mText.color = mOriginColor;

}

public void OnClick()

{

if (mView.mSelectedItem != null)

mView.mSelectedItem.OnHighLight(false);

mView.mSelectedItem = this;

OnHighLight(true);

}

public void OnScroll(PointerEventData eventData)

{

mView.mScrollRect.OnScroll(eventData);

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class UICategoryView : MonoBehaviour

{

public int mLayer;

public UICategoryItem mItemTemplate;

List<UICategoryItem> mItemList=new List<UICategoryItem>();

public UICategoryVO1 mVO { get; set; }

public UICategoryItem mSelectedItem { get; set; }

float uiTemplateHeight;

Vector2 contentSize;

Scrollbar mScrollbar;

[HideInInspector]

public ScrollRect mScrollRect;

int mMaxRow;

// Use this for initialization

void Awake ()

{

uiTemplateHeight = mItemTemplate.GetComponent<RectTransform>().rect.height+1;

contentSize = mItemTemplate.mView.GetComponent<RectTransform>().sizeDelta;

mMaxRow = (int)(contentSize.y/uiTemplateHeight);

mScrollbar = GetComponentInChildren<Scrollbar>();

mScrollRect = GetComponent<ScrollRect>();

}

private void Start()

{

ScrollToSelectedItem();

}

public void SetLayer(int ly)

{

mLayer = ly;

gameObject.name = "CategoryView " + ly;

}

public void SetVO(UICategoryVO1 vo)

{

mVO = vo;

ClearItems();

if (mVO.children != null)

{

GameObject itemGO;

UICategoryItem uiItem;

if (mVO.children.Count \* uiTemplateHeight > contentSize.y)

{

mItemTemplate.transform.parent.GetComponent<RectTransform>().sizeDelta =

new Vector2(contentSize.x, mVO.children.Count \* uiTemplateHeight+5);

}

for (int i=0;i<mVO.children.Count;i++)

{

itemGO = GameObject.Instantiate(mItemTemplate.gameObject,mItemTemplate.transform.parent);

itemGO.SetActive(true);

uiItem = itemGO.GetComponent<UICategoryItem>();

uiItem.SetVO(mVO.children[i]);

uiItem.SetIndex(i);

uiItem.transform.localPosition = mItemTemplate.transform.localPosition

+ Vector3.down \* i\* uiTemplateHeight;

mItemList.Add(uiItem);

}

}

}

public void ClearItems()

{

for(int i=mItemList.Count-1;i>=0;i--)

{

GameObject.Destroy(mItemList[i]);

}

mItemList.Clear();

}

public void SelectItemByName(string name)

{

mSelectedItem = null;

for(int i=0;i<mItemList.Count;i++)

{

if(mItemList[i].mText.text == name)

{

mSelectedItem = mItemList[i];

mSelectedItem.OnHighLight(true);

break;

}

}

}

public void ScrollToSelectedItem()

{

if (mSelectedItem != null)

{

if (mSelectedItem.mIndex >= mMaxRow)

{

mScrollbar.value =1-(float)(mSelectedItem.mIndex + 1 - mMaxRow) / (mItemList.Count - mMaxRow);

mScrollbar.value = Mathf.Max(0.0001f, mScrollbar.value);

}

}

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class UIDropdown : MonoBehaviour

{

[HideInInspector]

public Dropdown mDropDown;

public GameObject mPlaceHolder;

public string mPlayerPrefsType = "";

private void Awake()

{

mDropDown = GetComponent<Dropdown>();

}

private void OnEnable()

{

mDropDown.value = PlayerPrefs.GetInt(mPlayerPrefsType, 0);

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class UIGroupProductWin : UIBaseWin

{

public InputField mCode, mName, mType;

public UICategory mCategory;

public UIScrollView mScrollLeft, mScrollRight;

public UIGroupProductVO2 mGroupVO;

public Button mConfirmBtn;

[HideInInspector]

public bool mConfirmEnabled;

public Toggle mToggleSelectAll;

// Use this for initialization

void Awake()

{

mGroupVO = new UIGroupProductVO2();

//GetProductList();

}

private void OnEnable()

{

mConfirmEnabled = true;

OnBtnReset();

mToggleSelectAll.isOn = false;

GetProductList();

mScrollRight.Clear();

}

public void OnBtnSearch()

{

mGroupVO.Reset();

mGroupVO.productCode = mCode.text;

mGroupVO.productName = mName.text;

mGroupVO.modelNumber = mType.text;

if (mCategory.mCategoryCode != "")

{

string[] cc = mCategory.mCategoryCode.Split(',');

mGroupVO.productType = cc[cc.Length - 1];

}

GetProductList();

}

public void OnBtnReset()

{

mCode.text = "";

mName.text = "";

mType.text = "";

mGroupVO.Reset();

mCategory.Reset();

GetProductList();

}

public void OnBtnConfirm()

{

if (!mConfirmEnabled)

return;

string param = "";

foreach(UIScrollViewItem item in mScrollRight.mDic.Values)

param+=","+ item.mVO.modelId;

if (param != "")

param = param.Substring(1);

StartCoroutine(Net.Instance.GetGroupProductFile(param));

mConfirmEnabled = false;

}

public void ShowList()

{

mScrollLeft.AddItems(GroundMain.Instance.mProductList);

}

public void ClearList()

{

mScrollLeft.Clear();

//mScrollRight.Clear();

}

public void Reset()

{

mScrollLeft.Clear();

mScrollRight.Clear();

OnBtnReset();

}

public void GetProductList()

{

StartCoroutine(Net.Instance.GetGroupProductList(mGroupVO.ObjToString()));

}

public void OnBtnClose()

{

UIManager.Instance.Close("GroupProduct");

}

public void OnSelectItem(UIScrollViewItem item)

{

if(item.mToggle.isOn)

mScrollRight.AddItemAndResize(item.mVO);

}

public void OnRemoveItem(UIScrollViewItem item)

{

mScrollRight.RemoveItemAndResize(item);

foreach (KeyValuePair<int, UIScrollViewItem> kv in mScrollLeft.mDic)

{

if (kv.Key == item.mVO.id)

kv.Value.mToggle.isOn = false;

}

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class UILoginWin : UIBaseWin

{

public InputField nameInput,pwdInput;

public Button loginButton;

public Toggle rememberToggle;

public Text tipsText;

Dropdown selServer;

[HideInInspector]

public bool mLoginEnabled = true;

// Use this for initialization

private void Awake()

{

selServer = transform.Find("Dropdown").GetComponent<Dropdown>() ;

selServer.value = 0;

if (NetConst.Instance.GetUrl("SelectServer") == "1")

{

selServer.gameObject.SetActive(true);

if (PlayerPrefs.HasKey("SelServer"))

selServer.value = PlayerPrefs.GetInt("SelServer");

}

else

{

selServer.gameObject.SetActive(false);

selServer.value = 2;

}

OnServerChange();

}

private void OnEnable()

{

if (rememberToggle.isOn)

{

nameInput.text = PlayerPrefs.GetString("username");

pwdInput.text = PlayerPrefs.GetString("userpwd");

}

SetTips("");

}

public void OnBtnLogin()

{

if (!mLoginEnabled)

return;

#if UNITY\_EDITOR

DoLogin();

#else

if (NetConst.Instance.GetUrl("CheckVersion") == "1")

NetConst.Instance.StartCoroutine( NetConst.Instance.ReqVersion());

else

DoLogin();

#endif

if (rememberToggle.isOn)

{

PlayerPrefs.SetString("username", nameInput.text);

PlayerPrefs.SetString("userpwd", pwdInput.text);

}

else

{

PlayerPrefs.SetString("username", "");

PlayerPrefs.SetString("userpwd", "");

}

mLoginEnabled=false;

SetTips("");

}

public void DoLogin()

{

Net.Instance.StartCoroutine(Net.Instance.Login(nameInput.text, pwdInput.text));

}

public void SetTips(string tips)

{

tipsText.text = tips;

}

public void OnServerChange()

{

if (selServer.value == 0)

NetConst.Instance.mServerTag = "dev.";

else if (selServer.value == 1)

NetConst.Instance.mServerTag = "ci.";

else if (selServer.value == 2)

NetConst.Instance.mServerTag = "";

PlayerPrefs.SetInt("SelServer", selServer.value);

//Debug.Log("Net.Instance.mServerTag ="+ Net.Instance.mServerTag);

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using System.Runtime.InteropServices;

using System.IO;

public class UIProgressWin : UIBaseWin

{

public RectTransform progressBg, progressBar;

public Text percent;

float bgWidth;

private void Awake()

{

bgWidth = progressBg.rect.width;

}

private void Update()

{

if (Net.Instance.mUploadRequest == null)

{

progressBar.sizeDelta = new Vector2(0, progressBar.rect.height);

percent.text = "0% " +(NetConst.Instance.uploadModelIndex + 1) + "/" + NetConst.Instance.uploadModelTotal;

}

else

{

progressBar.sizeDelta = new Vector2(Net.Instance.mUploadRequest.uploadProgress \* progressBg.rect.width, progressBar.rect.height);

percent.text = (int)(Net.Instance.mUploadRequest.uploadProgress \* 100) + "% " +

(NetConst.Instance.uploadModelIndex + 1) + "/" + NetConst.Instance.uploadModelTotal;

}

}

public void SetProgress(float value)

{

progressBar.sizeDelta = new Vector2(value \* progressBg.rect.width, progressBar.rect.height);

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class UIScrollView : MonoBehaviour

{

public UIScrollViewItem mItem;

float mItemHeight;

Vector2 mContentSize;

public RectTransform mContentRect;

public Dictionary<int,UIScrollViewItem> mDic = new Dictionary<int, UIScrollViewItem>();

public Toggle mToggle;

// Use this for initialization

void Start ()

{

mItemHeight = mItem.GetComponent<RectTransform>().rect.height + 1;

mContentSize = mContentRect.sizeDelta;

}

public void AddItem(UIGroupProductVO1 productVO)

{

if (mDic.ContainsKey(productVO.id))

return;

GameObject itemGO = GameObject.Instantiate(mItem.gameObject,mItem.transform.parent);

itemGO.SetActive(true);

itemGO.transform.localPosition = mItem.transform.localPosition + Vector3.down \* mDic.Count\* mItemHeight;

itemGO.GetComponent<UIScrollViewItem>().SetVO(productVO);

mDic.Add(productVO.id,itemGO.GetComponent<UIScrollViewItem>());

}

public void AddItems(List<UIGroupProductVO1> list)

{

foreach (UIGroupProductVO1 vo in list)

{

AddItem(vo);

}

Resize();

}

public void AddItemAndResize(UIGroupProductVO1 productVO)

{

AddItem(productVO);

Resize();

}

public void Resize()

{

if (mDic.Count \* mItemHeight > mContentSize.y)

{

mContentRect.sizeDelta = new Vector2(mContentSize.x, mDic.Count \* mItemHeight + 5);

}

else

{

mContentRect.sizeDelta = mContentSize;

}

}

public void RemoveItem(UIScrollViewItem item)

{

mDic.Remove(item.mVO.id);

GameObject.Destroy(item.gameObject);

int i = 0;

foreach(UIScrollViewItem uiSVI in mDic.Values)

{

uiSVI.transform.localPosition = mItem.transform.localPosition + Vector3.down \* i \* mItemHeight;

i++;

}

}

public void RemoveItemAndResize(UIScrollViewItem item)

{

RemoveItem(item);

Resize();

}

public void Clear()

{

foreach(UIScrollViewItem item in mDic.Values)

{

GameObject.Destroy(item.gameObject);

}

mDic.Clear();

Resize();

}

public void OnBtnSelectAll()

{

foreach(UIScrollViewItem item in mDic.Values)

{

item.mToggle.isOn = mToggle.isOn;

}

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class UIScrollViewItem : MonoBehaviour

{

public UIScrollView mView;

public Image mImage;

public Text mCode, mName, mType;

public Toggle mToggle;

public UIGroupProductVO1 mVO { get; set; }

// Use this for initialization

void Start ()

{

}

public void SetVO(UIGroupProductVO1 vo)

{

mVO = vo;

mCode.text = mVO.productCode;

mName.text = mVO.productName;

mType.text = mVO.modelNumber;

if(vo.picPath != null && vo.picPath !="")

StartCoroutine(Net.Instance.DownLoadTexture(this, vo));

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

public class UIUploadGroupWin : UIBaseWin

{

public InputField modelNameInput;

public Text tipsText;

public Button cancelBtn, uploadBtn;

public UIDropdown uiBrandDropDown, uiGroupDropDown;

[HideInInspector]

public bool mUploadEnabled = true;

private void Start()

{

List<Dropdown.OptionData> brandlist = new List<Dropdown.OptionData>();

foreach (CommonType type in NetConst.Instance.mBrandTypeList)

brandlist.Add(new Dropdown.OptionData {text=type.name });

uiBrandDropDown.mDropDown.ClearOptions();

uiBrandDropDown.mDropDown.AddOptions(brandlist);

uiBrandDropDown.mPlayerPrefsType = "groupProductBrand";

List<Dropdown.OptionData> grouplist = new List<Dropdown.OptionData>();

foreach (CommonType type in NetConst.Instance.mProductTypeList)

grouplist.Add(new Dropdown.OptionData { text = type.name });

uiGroupDropDown.mDropDown.ClearOptions();

uiGroupDropDown.mDropDown.AddOptions(grouplist);

uiGroupDropDown.mPlayerPrefsType = "groupProductType";

}

private void OnEnable()

{

modelNameInput.text = PlayerPrefs.GetString("groupProductName");

mUploadEnabled = true;

tipsText.text = "";

}

public void OnBtnUpload()

{

if (!mUploadEnabled)

return;

tipsText.text = CheckBeforeUpload();

if (tipsText.text != "")

return;

GroupProductInfo mProductInfo = new GroupProductInfo();

mProductInfo.groupName = modelNameInput.text;

mProductInfo.groupType = NetConst.Instance.mProductTypeList[uiGroupDropDown.mDropDown.value].id;

mProductInfo.brandId = NetConst.Instance.mBrandTypeList[uiBrandDropDown.mDropDown.value].id; ;

StartCoroutine(Net.Instance.UploadGroupProduct(mProductInfo));

PlayerPrefs.SetString("groupProductName", modelNameInput.text);

PlayerPrefs.SetInt("groupProductType", uiGroupDropDown.mDropDown.value);

PlayerPrefs.SetInt("groupProductBrand", uiBrandDropDown.mDropDown.value);

mUploadEnabled = false;

}

protected string CheckBeforeUpload()

{

if (modelNameInput.text == "")

return "请填写模型名";

if (uiGroupDropDown.mDropDown.captionText.text == "")

return "请选择模型类型";

if (uiBrandDropDown.mDropDown.captionText.text == "")

return "请选择品牌";

if (GroundConstObject.mainProductID == -1)

return "请设置主产品";

if (GroundConstObject.mainProductID == -2)

return "请不要将重复的产品设置为主产品";

return "";

}

public void SetTips(string tips)

{

tipsText.text = tips;

}

public void OnBtnClose()

{

UIManager.Instance.Close("UploadGroup");

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

using UnityEngine.UI;

public class UIUploadHardWin : UIUploadWin

{

public InputField productInput;

public static UIUploadHardWin Instance;

private void Awake()

{

Instance = this;

Init();

}

protected override void Init()

{

UIName = "UploadHard";

}

private void OnDestroy()

{

Instance = null;

}

private void OnEnable()

{

if (PlayerPrefs.HasKey("modelNameHard"))

modelNameInput.text = PlayerPrefs.GetString("modelNameHard");

if (PlayerPrefs.HasKey("productModel"))

productInput.text = PlayerPrefs.GetString("productModel");

mUploadEnabled = true;

}

IEnumerator ExportHardModel()

{

Net.Instance.isUploading = true;

doTest.Instance.OutSideHardOpen(NetConst.Instance.uploadModelIndex);

yield return null;

ExportFbx();

}

private void Update()

{

if (NetConst.Instance.uploadModelIndex >= NetConst.Instance.uploadModelTotal-1)

return;

if(!Net.Instance.isUploading)

{

AppMain.Instance.ReleaseMemory();

}

}

public void ContinueUpload()

{

if (NetConst.Instance.uploadModelIndex >= NetConst.Instance.uploadModelTotal - 1)

return;

NetConst.Instance.uploadModelIndex++;

Net.Instance.mUploadRequest = null;

StartCoroutine(ExportHardModel());

}

protected override void CreateModelInfo()

{

base.CreateModelInfo();

PlayerPrefs.SetString("modelNameHard", modelNameInput.text);

Net.Instance.mModelInfo.modelNum = productInput.text;

PlayerPrefs.SetString("productModel", productInput.text);

Net.Instance.mModelInfo.mainModelFlag = NetConst.Instance.uploadModelIndex == HardConst.HardFbxId?1:0;

}

public override void OnBtnUpload()

{

NetConst.Instance.uploadModelTotal = HardConst.HardBatchLastFbxs.Count;

NetConst.Instance.uploadModelIndex = -1;

}

protected override string CheckBeforeUpload()

{

if (modelNameInput.text == "")

return "请填写模型名";

if (productInput.text == "")

return "请填写模型型号";

return "";

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using System.Runtime.InteropServices;

using System.IO;

public class UIUploadMatWin : UIUploadWin

{

public InputField texLengthInput, texWidthInput;

protected override void Init()

{

UIName = "UploadMat";

}

private void OnEnable()

{

if(PlayerPrefs.HasKey("materialName"))

modelNameInput.text = PlayerPrefs.GetString("materialName");

mUploadEnabled = true;

}

public override void OnBtnUpload()

{

if (!mUploadEnabled)

return;

tipsText.text = CheckBeforeUpload();

if (tipsText.text != "")

return;

if (File.Exists(NetConst.Instance.tduFileName))

File.Delete(NetConst.Instance.tduFileName);

SaveFbxFile.mSaveType = SaveType.SINGLE\_MAT;

SaveFbxFile.Instance.SaveMatLocal(MyFBXFileConst.toggleMat, MyFBXFileConst.ToggleIndex);

Net.Instance.ClearJpgFolder();

Net.Instance.CreateMaterialTexture(MyFBXFileConst.toggleMat,"ball");

string sourceFolderLoad = MyFBXFileConst.FbxMatsName + "/" + MyFBXFileConst.ToggleIndex;

ZipTools.ZipHelper.Zip(sourceFolderLoad, NetConst.Instance.tduFileName);

ModelInfo mModelInfo = new ModelInfo();

mModelInfo.modelName = modelNameInput.text;

if (texWidthInput.text != "")

mModelInfo.length = int.Parse(texWidthInput.text);

if (texLengthInput.text != "")

mModelInfo.width = int.Parse(texLengthInput.text);

PlayerPrefs.SetString("materialName", modelNameInput.text);

StartCoroutine(Net.Instance.PostTexture(mModelInfo,"ball"));

mUploadEnabled = false;

}

protected override string CheckBeforeUpload()

{

if (modelNameInput.text == "")

return "请填写模型名";

if (MyFBXFileConst.toggleMat == null)

return "请先选择材质";

return "";

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

using UnityEngine.UI;

public class UIUploadResultWin : UIBaseWin

{

public Text tipsText;

public void SetTips(string tips,bool error = false)

{

if (error)

tipsText.color = Color.red;

else

tipsText.color = Color.white;

tipsText.text = tips;

}

public void CloseWin()

{

UIManager.Instance.Close("UploadResult");

NetConst.Instance.uploadModelIndex = 0;

NetConst.Instance.uploadModelTotal = 0;

if (SceneManager.GetActiveScene().name.Equals("HardScene"))

{

AppMain.Instance.ReleaseMemory();

}

}

}

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using System.Runtime.InteropServices;

using System.IO;

using UnityEngine.SceneManagement;

public class UIUploadWin : UIBaseWin

{

public InputField modelNameInput;

public Text tipsText;

public Button cancelBtn,uploadBtn;

public UICategory uiCategory;

[HideInInspector]

public bool mUploadEnabled=true;

protected string UIName;

private void Awake()

{

Init();

}

protected virtual void Init()

{

UIName = "Upload";

}

private void OnEnable()

{

if (PlayerPrefs.HasKey("modelName"))

modelNameInput.text = PlayerPrefs.GetString("modelName");

mUploadEnabled = true;

}

public string GetCategoryCode(List<UICategoryVO1> list,string[] names,int index)

{

foreach (UICategoryVO1 vo in list)

{

if (vo.name == names[index])

{

if (vo.children == null)

return vo.keyCode;

else

return vo.keyCode + "," + GetCategoryCode(vo.children, names, index + 1);

}

}

return "";

}

private void OnDisable()

{

modelNameInput.text = "";

tipsText.text = "";

}

protected void ExportFbx()

{

if (MyFBXFileConst.mainGameObj == null)

return;

else

doTest.Instance.ResetPos();

if(Directory.GetFiles(MyFBXFileConst.TwoInOneMatName, "\*.fbx").Length != 1)

{

UIManager.Instance.Close("Upload");

UIManager.Instance.Open("UploadResult");

(UIManager.Instance.GetWin("UploadResult") as UIUploadResultWin).SetTips("没有找到FBX文件", true);

return;

}

Net.Instance.ClearModelInfo();

Net.Instance.ClearJpgFolder();

if (File.Exists(NetConst.Instance.tduFileName))

File.Delete(NetConst.Instance.tduFileName);

if (Directory.Exists(NetConst.Instance.tduFolder))

MyFBXFileConst.DeleteFolder(NetConst.Instance.tduFolder);

Directory.CreateDirectory(NetConst.Instance.tduFolder);

SaveFbxFile.Instance.SaveFbxLocal();

ZipTools.ZipHelper.Zip(MyFBXFileConst.TwoInOneMatName, NetConst.Instance.tduFileName);

//3DU文件大于50M不让上传

FileStream fs3DU = new FileStream(NetConst.Instance.tduFileName, FileMode.Open);

int fsLen = (int)fs3DU.Length;

if (fsLen > 1048576 \* 50)

{

UIManager.Instance.Close("Upload");

UIManager.Instance.Open("UploadResult");

(UIManager.Instance.GetWin("UploadResult") as UIUploadResultWin).SetTips("模型上传失败:文件过大", true);

fs3DU.Flush();

fs3DU.Close();

fs3DU.Dispose();

return;

}

fs3DU.Flush();

fs3DU.Close();

fs3DU.Dispose();

CreateModelInfo();

FbxAndMatInfo fbxInfo = new FbxAndMatInfo();

fbxInfo.index = 0;

fbxInfo.modelInfo = Net.Instance.mModelInfo;

fbxInfo.tduPath = NetConst.Instance.tduFileName;

Net.Instance.mFbxAndMatInfoList.Add(fbxInfo);

StartCoroutine(Net.Instance.PostTexture(fbxInfo.modelInfo, "model"));

if (Directory.Exists(MyFBXFileConst.FbxMatsName))

MyFBXFileConst.DeleteFolder(MyFBXFileConst.FbxMatsName);

Renderer render = MyFBXFileConst.mainGameObj.GetComponent<Renderer>();

SaveFbxFile.mSaveType = SaveType.MULTI\_MAT;

Material mat;

MatArea area;

string path;

for (int i = 0; i < MyFBXFileConst.mainGameMats.Length; i++)

{

mat = MyFBXFileConst.mainGameMats[i];

area = MyFBXFileConst.ListMat[i];

path = MyFBXFileConst.FbxMatsName + "/" + area.code;

SaveFbxFile.Instance.SaveMatLocal(mat, area.code);

Net.Instance.CreateMaterialTexture(mat, "model");

ZipTools.ZipHelper.Zip(path, NetConst.Instance.tduFolder + area.code + ".3DU");

ModelInfo modelInfo = new ModelInfo();

modelInfo.modelName = "" + area.code;

FbxAndMatInfo matInfo = new FbxAndMatInfo();

if(HardConst.isHard)

matInfo.index = area.code+1;

else

matInfo.index = i + 1;

matInfo.modelInfo = modelInfo;

matInfo.tduPath = NetConst.Instance.tduFolder + area.code + ".3DU";

Net.Instance.mFbxAndMatInfoList.Add(matInfo);

StartCoroutine(Net.Instance.PostTexture(modelInfo, "model"));

}

}

/// <summary>

/// 模型的数据

/// </summary>

protected virtual void CreateModelInfo()

{

Net.Instance.mModelInfo = new ModelInfo();

Net.Instance.mModelInfo.modelName = modelNameInput.text;

Net.Instance.mModelInfo.areas = MyFBXFileConst.ListMat;

string[] cc = uiCategory.mCategoryCode.Split(',');

Net.Instance.mModelInfo.categoryIds = cc[cc.Length - 1];

Net.Instance.mModelInfo.smallType = cc[cc.Length - 1];

//Net.Instance.mModelInfo.categoryIds = uiCategory.mCategoryCode;

//Net.Instance.mModelInfo.smallType = uiCategory.mCategoryCode;

Net.Instance.mModelInfo.length = Mathf.RoundToInt(MyFBXFileConst.LWH.x \* 100);

Net.Instance.mModelInfo.width = Mathf.RoundToInt(MyFBXFileConst.LWH.y \* 100);

Net.Instance.mModelInfo.height = Mathf.RoundToInt(MyFBXFileConst.LWH.z \* 100);

if (!SceneManager.GetActiveScene().name.Equals("HardScene"))

PlayerPrefs.SetString("modelName", modelNameInput.text);

}

/// <summary>

/// 保存3du文件并上传

/// </summary>

public virtual void OnBtnUpload()

{

if (!mUploadEnabled)

return;

tipsText.text = CheckBeforeUpload();

if (tipsText.text != "")

return;

NetConst.Instance.uploadModelTotal = 1;

NetConst.Instance.uploadModelIndex = 0;

ExportFbx();

mUploadEnabled = false;

}

/// <summary>

/// 上传3du文件前检查

/// </summary>

/// <returns>错误提示，没有错误则返回""</returns>

protected virtual string CheckBeforeUpload()

{

if (modelNameInput.text == "")

return "请填写模型名";

if (uiCategory.mCategoryText.text == "")

return "请选择模型类型";

if (MyFBXFileConst.mainGameObj == null)

return "请先导入fbx文件";

return "";

}

public void SetTips(string tips)

{

tipsText.text = tips;

}

public void OnBtnClose()

{

if (!mUploadEnabled)

return;

UIManager.Instance.Close(UIName);

}

}

package com.sandu.api.resmodel.model.bo;

import io.swagger.annotations.ApiModelProperty;

import java.io.Serializable;

import java.util.Date;

import java.util.List;

public class ModelListBO implements Serializable {

@ApiModelProperty("队形产品id")

private Integer concerProductId;

@ApiModelProperty("对应产品名称")

private String concerProductName;

@ApiModelProperty("对应产品名称")

private String concerProductCode;

@ApiModelProperty("id")

private Long id;

@ApiModelProperty("编码")

private String modelCode;

@ApiModelProperty("名称")

private String modelName;

@ApiModelProperty("缩略图地址")

private String thumbPicPath;

@ApiModelProperty("类型ids")

private String categoryIds;

@ApiModelProperty("类型")

private String categoryNames;

@ApiModelProperty("转化状态")

private String transStatus;

@ApiModelProperty("创建时间")

private Date gmtModified;

@ApiModelProperty("模型描述")

private String modelDesc;

@ApiModelProperty("模型文件地址")

private String modelPath;

@ApiModelProperty("模型长")

private Integer length;

@ApiModelProperty("模型宽")

private Integer width;

@ApiModelProperty("模型高")

private Integer height;

@ApiModelProperty("作者")

private String author;

@ApiModelProperty("模型来源")

private String originId;

@ApiModelProperty("模型来源公司的名称")

private String modelOrigin;

@ApiModelProperty("模型型号")

private String modelModelNum;

@ApiModelProperty("模型小类")

private String smallTypeMark;

@ApiModelProperty("模型大类")

private String TypeMark;

@ApiModelProperty("模型各区域贴图详情")

private List<ModelAreaBO> modelTextureInfo;

@ApiModelProperty("模型材质数量")

private Integer textureCount;

@ApiModelProperty("主模型标志 1:主模型,0:非主模型")

private Integer mainModelFlag;

public ModelListBO() {

}

public Integer getConcerProductId() {

return this.concerProductId;

}

public String getConcerProductName() {

return this.concerProductName;

}

public String getConcerProductCode() {

return this.concerProductCode;

}

public Long getId() {

return this.id;

}

public String getModelCode() {

return this.modelCode;

}

public String getModelName() {

return this.modelName;

}

public String getThumbPicPath() {

return this.thumbPicPath;

}

public String getCategoryIds() {

return this.categoryIds;

}

public String getCategoryNames() {

return this.categoryNames;

}

public String getTransStatus() {

return this.transStatus;

}

public Date getGmtModified() {

return this.gmtModified;

}

public String getModelDesc() {

return this.modelDesc;

}

public String getModelPath() {

return this.modelPath;

}

public Integer getLength() {

return this.length;

}

public Integer getWidth() {

return this.width;

}

public Integer getHeight() {

return this.height;

}

public String getAuthor() {

return this.author;

}

public String getOriginId() {

return this.originId;

}

public String getModelOrigin() {

return this.modelOrigin;

}

public String getModelModelNum() {

return this.modelModelNum;

}

public String getSmallTypeMark() {

return this.smallTypeMark;

}

public String getTypeMark() {

return this.TypeMark;

}

public List<ModelAreaBO> getModelTextureInfo() {

return this.modelTextureInfo;

}

public Integer getTextureCount() {

return this.textureCount;

}

public Integer getMainModelFlag() {

return this.mainModelFlag;

}

public void setConcerProductId(Integer concerProductId) {

this.concerProductId = concerProductId;

}

public void setConcerProductName(String concerProductName) {

this.concerProductName = concerProductName;

}

public void setConcerProductCode(String concerProductCode) {

this.concerProductCode = concerProductCode;

}

public void setId(Long id) {

this.id = id;

}

public void setModelCode(String modelCode) {

this.modelCode = modelCode;

}

public void setModelName(String modelName) {

this.modelName = modelName;

}

public void setThumbPicPath(String thumbPicPath) {

this.thumbPicPath = thumbPicPath;

}

public void setCategoryIds(String categoryIds) {

this.categoryIds = categoryIds;

}

public void setCategoryNames(String categoryNames) {

this.categoryNames = categoryNames;

}

public void setTransStatus(String transStatus) {

this.transStatus = transStatus;

}

public void setGmtModified(Date gmtModified) {

this.gmtModified = gmtModified;

}

public void setModelDesc(String modelDesc) {

this.modelDesc = modelDesc;

}

public void setModelPath(String modelPath) {

this.modelPath = modelPath;

}

public void setLength(Integer length) {

this.length = length;

}

public void setWidth(Integer width) {

this.width = width;

}

public void setHeight(Integer height) {

this.height = height;

}

public void setAuthor(String author) {

this.author = author;

}

public void setOriginId(String originId) {

this.originId = originId;

}

public void setModelOrigin(String modelOrigin) {

this.modelOrigin = modelOrigin;

}

public void setModelModelNum(String modelModelNum) {

this.modelModelNum = modelModelNum;

}

public void setSmallTypeMark(String smallTypeMark) {

this.smallTypeMark = smallTypeMark;

}

public void setTypeMark(String TypeMark) {

this.TypeMark = TypeMark;

}

public void setModelTextureInfo(List<ModelAreaBO> modelTextureInfo) {

this.modelTextureInfo = modelTextureInfo;

}

public void setTextureCount(Integer textureCount) {

this.textureCount = textureCount;

}

public void setMainModelFlag(Integer mainModelFlag) {

this.mainModelFlag = mainModelFlag;

}

public boolean equals(Object o) {

if (o == this) {

return true;

} else if (!(o instanceof ModelListBO)) {

return false;

} else {

ModelListBO other = (ModelListBO)o;

if (!other.canEqual(this)) {

return false;

} else {

label311: {

Object this$concerProductId = this.getConcerProductId();

Object other$concerProductId = other.getConcerProductId();

if (this$concerProductId == null) {

if (other$concerProductId == null) {

break label311;

}

} else if (this$concerProductId.equals(other$concerProductId)) {

break label311;

}

return false;

}

Object this$concerProductName = this.getConcerProductName();

Object other$concerProductName = other.getConcerProductName();

if (this$concerProductName == null) {

if (other$concerProductName != null) {

return false;

}

} else if (!this$concerProductName.equals(other$concerProductName)) {

return false;

}

label297: {

Object this$concerProductCode = this.getConcerProductCode();

Object other$concerProductCode = other.getConcerProductCode();

if (this$concerProductCode == null) {

if (other$concerProductCode == null) {

break label297;

}

} else if (this$concerProductCode.equals(other$concerProductCode)) {

break label297;

}

return false;

}

Object this$id = this.getId();

Object other$id = other.getId();

if (this$id == null) {

if (other$id != null) {

return false;

}

} else if (!this$id.equals(other$id)) {

return false;

}

label283: {

Object this$modelCode = this.getModelCode();

Object other$modelCode = other.getModelCode();

if (this$modelCode == null) {

if (other$modelCode == null) {

break label283;

}

} else if (this$modelCode.equals(other$modelCode)) {

break label283;

}

return false;

}

Object this$modelName = this.getModelName();

Object other$modelName = other.getModelName();

if (this$modelName == null) {

if (other$modelName != null) {

return false;

}

} else if (!this$modelName.equals(other$modelName)) {

return false;

}

label269: {

Object this$thumbPicPath = this.getThumbPicPath();

Object other$thumbPicPath = other.getThumbPicPath();

if (this$thumbPicPath == null) {

if (other$thumbPicPath == null) {

break label269;

}

} else if (this$thumbPicPath.equals(other$thumbPicPath)) {

break label269;

}

return false;

}

label262: {

Object this$categoryIds = this.getCategoryIds();

Object other$categoryIds = other.getCategoryIds();

if (this$categoryIds == null) {

if (other$categoryIds == null) {

break label262;

}

} else if (this$categoryIds.equals(other$categoryIds)) {

break label262;

}

return false;

}

Object this$categoryNames = this.getCategoryNames();

Object other$categoryNames = other.getCategoryNames();

if (this$categoryNames == null) {

if (other$categoryNames != null) {

return false;

}

} else if (!this$categoryNames.equals(other$categoryNames)) {

return false;

}

label248: {

Object this$transStatus = this.getTransStatus();

Object other$transStatus = other.getTransStatus();

if (this$transStatus == null) {

if (other$transStatus == null) {

break label248;

}

} else if (this$transStatus.equals(other$transStatus)) {

break label248;

}

return false;

}

label241: {

Object this$gmtModified = this.getGmtModified();

Object other$gmtModified = other.getGmtModified();

if (this$gmtModified == null) {

if (other$gmtModified == null) {

break label241;

}

} else if (this$gmtModified.equals(other$gmtModified)) {

break label241;

}

return false;

}

Object this$modelDesc = this.getModelDesc();

Object other$modelDesc = other.getModelDesc();

if (this$modelDesc == null) {

if (other$modelDesc != null) {

return false;

}

} else if (!this$modelDesc.equals(other$modelDesc)) {

return false;

}

Object this$modelPath = this.getModelPath();

Object other$modelPath = other.getModelPath();

if (this$modelPath == null) {

if (other$modelPath != null) {

return false;

}

} else if (!this$modelPath.equals(other$modelPath)) {

return false;

}

label220: {

Object this$length = this.getLength();

Object other$length = other.getLength();

if (this$length == null) {

if (other$length == null) {

break label220;

}

} else if (this$length.equals(other$length)) {

break label220;

}

return false;

}

Object this$width = this.getWidth();

Object other$width = other.getWidth();

if (this$width == null) {

if (other$width != null) {

return false;

}

} else if (!this$width.equals(other$width)) {

return false;

}

Object this$height = this.getHeight();

Object other$height = other.getHeight();

if (this$height == null) {

if (other$height != null) {

return false;

}

} else if (!this$height.equals(other$height)) {

return false;

}

label199: {

Object this$author = this.getAuthor();

Object other$author = other.getAuthor();

if (this$author == null) {

if (other$author == null) {

break label199;

}

} else if (this$author.equals(other$author)) {

break label199;

}

return false;

}

Object this$originId = this.getOriginId();

Object other$originId = other.getOriginId();

if (this$originId == null) {

if (other$originId != null) {

return false;

}

} else if (!this$originId.equals(other$originId)) {

return false;

}

label185: {

Object this$modelOrigin = this.getModelOrigin();

Object other$modelOrigin = other.getModelOrigin();

if (this$modelOrigin == null) {

if (other$modelOrigin == null) {

break label185;

}

} else if (this$modelOrigin.equals(other$modelOrigin)) {

break label185;

}

return false;

}

Object this$modelModelNum = this.getModelModelNum();

Object other$modelModelNum = other.getModelModelNum();

if (this$modelModelNum == null) {

if (other$modelModelNum != null) {

return false;

}

} else if (!this$modelModelNum.equals(other$modelModelNum)) {

return false;

}

label171: {

Object this$smallTypeMark = this.getSmallTypeMark();

Object other$smallTypeMark = other.getSmallTypeMark();

if (this$smallTypeMark == null) {

if (other$smallTypeMark == null) {

break label171;

}

} else if (this$smallTypeMark.equals(other$smallTypeMark)) {

break label171;

}

return false;

}

Object this$TypeMark = this.getTypeMark();

Object other$TypeMark = other.getTypeMark();

if (this$TypeMark == null) {

if (other$TypeMark != null) {

return false;

}

} else if (!this$TypeMark.equals(other$TypeMark)) {

return false;

}

label157: {

Object this$modelTextureInfo = this.getModelTextureInfo();

Object other$modelTextureInfo = other.getModelTextureInfo();

if (this$modelTextureInfo == null) {

if (other$modelTextureInfo == null) {

break label157;

}

} else if (this$modelTextureInfo.equals(other$modelTextureInfo)) {

break label157;

}

return false;

}

label150: {

Object this$textureCount = this.getTextureCount();

Object other$textureCount = other.getTextureCount();

if (this$textureCount == null) {

if (other$textureCount == null) {

break label150;

}

} else if (this$textureCount.equals(other$textureCount)) {

break label150;

}

return false;

}

Object this$mainModelFlag = this.getMainModelFlag();

Object other$mainModelFlag = other.getMainModelFlag();

if (this$mainModelFlag == null) {

if (other$mainModelFlag != null) {

return false;

}

} else if (!this$mainModelFlag.equals(other$mainModelFlag)) {

return false;

}

return true;

}

}

}

protected boolean canEqual(Object other) {

return other instanceof ModelListBO;

}

public int hashCode() {

int PRIME = true;

int result = 1;

Object $concerProductId = this.getConcerProductId();

int result = result \* 59 + ($concerProductId == null ? 43 : $concerProductId.hashCode());

Object $concerProductName = this.getConcerProductName();

result = result \* 59 + ($concerProductName == null ? 43 : $concerProductName.hashCode());

Object $concerProductCode = this.getConcerProductCode();

result = result \* 59 + ($concerProductCode == null ? 43 : $concerProductCode.hashCode());

Object $id = this.getId();

result = result \* 59 + ($id == null ? 43 : $id.hashCode());

Object $modelCode = this.getModelCode();

result = result \* 59 + ($modelCode == null ? 43 : $modelCode.hashCode());

Object $modelName = this.getModelName();

result = result \* 59 + ($modelName == null ? 43 : $modelName.hashCode());

Object $thumbPicPath = this.getThumbPicPath();

result = result \* 59 + ($thumbPicPath == null ? 43 : $thumbPicPath.hashCode());

Object $categoryIds = this.getCategoryIds();

result = result \* 59 + ($categoryIds == null ? 43 : $categoryIds.hashCode());

Object $categoryNames = this.getCategoryNames();

result = result \* 59 + ($categoryNames == null ? 43 : $categoryNames.hashCode());

Object $transStatus = this.getTransStatus();

result = result \* 59 + ($transStatus == null ? 43 : $transStatus.hashCode());

Object $gmtModified = this.getGmtModified();

result = result \* 59 + ($gmtModified == null ? 43 : $gmtModified.hashCode());

Object $modelDesc = this.getModelDesc();

result = result \* 59 + ($modelDesc == null ? 43 : $modelDesc.hashCode());

Object $modelPath = this.getModelPath();

result = result \* 59 + ($modelPath == null ? 43 : $modelPath.hashCode());

Object $length = this.getLength();

result = result \* 59 + ($length == null ? 43 : $length.hashCode());

Object $width = this.getWidth();

result = result \* 59 + ($width == null ? 43 : $width.hashCode());

Object $height = this.getHeight();

result = result \* 59 + ($height == null ? 43 : $height.hashCode());

Object $author = this.getAuthor();

result = result \* 59 + ($author == null ? 43 : $author.hashCode());

Object $originId = this.getOriginId();

result = result \* 59 + ($originId == null ? 43 : $originId.hashCode());

Object $modelOrigin = this.getModelOrigin();

result = result \* 59 + ($modelOrigin == null ? 43 : $modelOrigin.hashCode());

Object $modelModelNum = this.getModelModelNum();

result = result \* 59 + ($modelModelNum == null ? 43 : $modelModelNum.hashCode());

Object $smallTypeMark = this.getSmallTypeMark();

result = result \* 59 + ($smallTypeMark == null ? 43 : $smallTypeMark.hashCode());

Object $TypeMark = this.getTypeMark();

result = result \* 59 + ($TypeMark == null ? 43 : $TypeMark.hashCode());

Object $modelTextureInfo = this.getModelTextureInfo();

result = result \* 59 + ($modelTextureInfo == null ? 43 : $modelTextureInfo.hashCode());

Object $textureCount = this.getTextureCount();

result = result \* 59 + ($textureCount == null ? 43 : $textureCount.hashCode());

Object $mainModelFlag = this.getMainModelFlag();

result = result \* 59 + ($mainModelFlag == null ? 43 : $mainModelFlag.hashCode());

return result;

}

public String toString() {

return "ModelListBO(concerProductId=" + this.getConcerProductId() + ", concerProductName=" + this.getConcerProductName() + ", concerProductCode=" + this.getConcerProductCode() + ", id=" + this.getId() + ", modelCode=" + this.getModelCode() + ", modelName=" + this.getModelName() + ", thumbPicPath=" + this.getThumbPicPath() + ", categoryIds=" + this.getCategoryIds() + ", categoryNames=" + this.getCategoryNames() + ", transStatus=" + this.getTransStatus() + ", gmtModified=" + this.getGmtModified() + ", modelDesc=" + this.getModelDesc() + ", modelPath=" + this.getModelPath() + ", length=" + this.getLength() + ", width=" + this.getWidth() + ", height=" + this.getHeight() + ", author=" + this.getAuthor() + ", originId=" + this.getOriginId() + ", modelOrigin=" + this.getModelOrigin() + ", modelModelNum=" + this.getModelModelNum() + ", smallTypeMark=" + this.getSmallTypeMark() + ", TypeMark=" + this.getTypeMark() + ", modelTextureInfo=" + this.getModelTextureInfo() + ", textureCount=" + this.getTextureCount() + ", mainModelFlag=" + this.getMainModelFlag() + ")";

}

}

//

// Source code recreated from a .class file by IntelliJ IDEA

// (powered by Fernflower decompiler)

//

package com.sandu.api.resmodel.model.bo;

import io.swagger.annotations.ApiModelProperty;

import java.io.Serializable;

import java.math.BigDecimal;

import java.util.Date;

import java.util.List;

public class ModelBO implements Serializable {

private Long id;

private String modelCode;

private String modelName;

private String modelFileName;

private String modelType;

private String modelSize;

private String modelSuffix;

private String modelLevel;

private String modelPath;

private String modelDesc;

private String sysCode;

private String creator;

private Date gmtCreate;

private String modifier;

private Date gmtModified;

private Integer isDeleted;

private Integer modelOrdering;

private String fileKey;

private String fileKeys;

private String businessIds;

private String att4;

private String att5;

private String att6;

private Date dateAtt1;

private Date dateAtt2;

private Integer businessId;

private Integer numAtt2;

private BigDecimal numAtt3;

private BigDecimal numAtt4;

private String remark;

private Integer length;

private Integer height;

private Integer width;

private Integer minHeight;

private Integer isModelShare;

private Integer isDecompress;

private String thumbPicPath;

private String categoryIds;

private String categoryNames;

private String transStatus;

private String modelModelNum;

private Integer companyId;

private String file3duPath;

private Integer productId;

private Long originId;

private String typeMark;

private String smallTypeMark;

private List<String> typeNames;

private List<ModelAreaBO> modelTextureInfo;

@ApiModelProperty("模型材质数量")

private Integer textureCount;

@ApiModelProperty("主模型标志 true:为主模型,flse:副模型 ")

private Boolean mainProductFlag;

public ModelBO() {

}

public Long getId() {

return this.id;

}

public String getModelCode() {

return this.modelCode;

}

public String getModelName() {

return this.modelName;

}

public String getModelFileName() {

return this.modelFileName;

}

public String getModelType() {

return this.modelType;

}

public String getModelSize() {

return this.modelSize;

}

public String getModelSuffix() {

return this.modelSuffix;

}

public String getModelLevel() {

return this.modelLevel;

}

public String getModelPath() {

return this.modelPath;

}

public String getModelDesc() {

return this.modelDesc;

}

public String getSysCode() {

return this.sysCode;

}

public String getCreator() {

return this.creator;

}

public Date getGmtCreate() {

return this.gmtCreate;

}

public String getModifier() {

return this.modifier;

}

public Date getGmtModified() {

return this.gmtModified;

}

public Integer getIsDeleted() {

return this.isDeleted;

}

public Integer getModelOrdering() {

return this.modelOrdering;

}

public String getFileKey() {

return this.fileKey;

}

public String getFileKeys() {

return this.fileKeys;

}

public String getBusinessIds() {

return this.businessIds;

}

public String getAtt4() {

return this.att4;

}

public String getAtt5() {

return this.att5;

}

public String getAtt6() {

return this.att6;

}

public Date getDateAtt1() {

return this.dateAtt1;

}

public Date getDateAtt2() {

return this.dateAtt2;

}

public Integer getBusinessId() {

return this.businessId;

}

public Integer getNumAtt2() {

return this.numAtt2;

}

public BigDecimal getNumAtt3() {

return this.numAtt3;

}

public BigDecimal getNumAtt4() {

return this.numAtt4;

}

public String getRemark() {

return this.remark;

}

public Integer getLength() {

return this.length;

}

public Integer getHeight() {

return this.height;

}

public Integer getWidth() {

return this.width;

}

public Integer getMinHeight() {

return this.minHeight;

}

public Integer getIsModelShare() {

return this.isModelShare;

}

public Integer getIsDecompress() {

return this.isDecompress;

}

public String getThumbPicPath() {

return this.thumbPicPath;

}

public String getCategoryIds() {

return this.categoryIds;

}

public String getCategoryNames() {

return this.categoryNames;

}

public String getTransStatus() {

return this.transStatus;

}

public String getModelModelNum() {

return this.modelModelNum;

}

public Integer getCompanyId() {

return this.companyId;

}

public String getFile3duPath() {

return this.file3duPath;

}

public Integer getProductId() {

return this.productId;

}

public Long getOriginId() {

return this.originId;

}

public String getTypeMark() {

return this.typeMark;

}

public String getSmallTypeMark() {

return this.smallTypeMark;

}

public List<String> getTypeNames() {

return this.typeNames;

}

public List<ModelAreaBO> getModelTextureInfo() {

return this.modelTextureInfo;

}

public Integer getTextureCount() {

return this.textureCount;

}

public Boolean getMainProductFlag() {

return this.mainProductFlag;

}

public void setId(Long id) {

this.id = id;

}

public void setModelCode(String modelCode) {

this.modelCode = modelCode;

}

public void setModelName(String modelName) {

this.modelName = modelName;

}

public void setModelFileName(String modelFileName) {

this.modelFileName = modelFileName;

}

public void setModelType(String modelType) {

this.modelType = modelType;

}

public void setModelSize(String modelSize) {

this.modelSize = modelSize;

}

public void setModelSuffix(String modelSuffix) {

this.modelSuffix = modelSuffix;

}

public void setModelLevel(String modelLevel) {

this.modelLevel = modelLevel;

}

public void setModelPath(String modelPath) {

this.modelPath = modelPath;

}

public void setModelDesc(String modelDesc) {

this.modelDesc = modelDesc;

}

public void setSysCode(String sysCode) {

this.sysCode = sysCode;

}

public void setCreator(String creator) {

this.creator = creator;

}

public void setGmtCreate(Date gmtCreate) {

this.gmtCreate = gmtCreate;

}

public void setModifier(String modifier) {

this.modifier = modifier;

}

public void setGmtModified(Date gmtModified) {

this.gmtModified = gmtModified;

}

public void setIsDeleted(Integer isDeleted) {

this.isDeleted = isDeleted;

}

public void setModelOrdering(Integer modelOrdering) {

this.modelOrdering = modelOrdering;

}

public void setFileKey(String fileKey) {

this.fileKey = fileKey;

}

public void setFileKeys(String fileKeys) {

this.fileKeys = fileKeys;

}

public void setBusinessIds(String businessIds) {

this.businessIds = businessIds;

}

public void setAtt4(String att4) {

this.att4 = att4;

}

public void setAtt5(String att5) {

this.att5 = att5;

}

public void setAtt6(String att6) {

this.att6 = att6;

}

public void setDateAtt1(Date dateAtt1) {

this.dateAtt1 = dateAtt1;

}

public void setDateAtt2(Date dateAtt2) {

this.dateAtt2 = dateAtt2;

}

public void setBusinessId(Integer businessId) {

this.businessId = businessId;

}

public void setNumAtt2(Integer numAtt2) {

this.numAtt2 = numAtt2;

}

public void setNumAtt3(BigDecimal numAtt3) {

this.numAtt3 = numAtt3;

}

public void setNumAtt4(BigDecimal numAtt4) {

this.numAtt4 = numAtt4;

}

public void setRemark(String remark) {

this.remark = remark;

}

public void setLength(Integer length) {

this.length = length;

}

public void setHeight(Integer height) {

this.height = height;

}

public void setWidth(Integer width) {

this.width = width;

}

public void setMinHeight(Integer minHeight) {

this.minHeight = minHeight;

}

public void setIsModelShare(Integer isModelShare) {

this.isModelShare = isModelShare;

}

public void setIsDecompress(Integer isDecompress) {

this.isDecompress = isDecompress;

}

public void setThumbPicPath(String thumbPicPath) {

this.thumbPicPath = thumbPicPath;

}

public void setCategoryIds(String categoryIds) {

this.categoryIds = categoryIds;

}

public void setCategoryNames(String categoryNames) {

this.categoryNames = categoryNames;

}

public void setTransStatus(String transStatus) {

this.transStatus = transStatus;

}

public void setModelModelNum(String modelModelNum) {

this.modelModelNum = modelModelNum;

}

public void setCompanyId(Integer companyId) {

this.companyId = companyId;

}

public void setFile3duPath(String file3duPath) {

this.file3duPath = file3duPath;

}

public void setProductId(Integer productId) {

this.productId = productId;

}

public void setOriginId(Long originId) {

this.originId = originId;

}

public void setTypeMark(String typeMark) {

this.typeMark = typeMark;

}

public void setSmallTypeMark(String smallTypeMark) {

this.smallTypeMark = smallTypeMark;

}

public void setTypeNames(List<String> typeNames) {

this.typeNames = typeNames;

}

public void setModelTextureInfo(List<ModelAreaBO> modelTextureInfo) {

this.modelTextureInfo = modelTextureInfo;

}

public void setTextureCount(Integer textureCount) {

this.textureCount = textureCount;

}

public void setMainProductFlag(Boolean mainProductFlag) {

this.mainProductFlag = mainProductFlag;

}

public boolean equals(Object o) {

if (o == this) {

return true;

} else if (!(o instanceof ModelBO)) {

return false;

} else {

ModelBO other = (ModelBO)o;

if (!other.canEqual(this)) {

return false;

} else {

label623: {

Object this$id = this.getId();

Object other$id = other.getId();

if (this$id == null) {

if (other$id == null) {

break label623;

}

} else if (this$id.equals(other$id)) {

break label623;

}

return false;

}

Object this$modelCode = this.getModelCode();

Object other$modelCode = other.getModelCode();

if (this$modelCode == null) {

if (other$modelCode != null) {

return false;

}

} else if (!this$modelCode.equals(other$modelCode)) {

return false;

}

Object this$modelName = this.getModelName();

Object other$modelName = other.getModelName();

if (this$modelName == null) {

if (other$modelName != null) {

return false;

}

} else if (!this$modelName.equals(other$modelName)) {

return false;

}

label602: {

Object this$modelFileName = this.getModelFileName();

Object other$modelFileName = other.getModelFileName();

if (this$modelFileName == null) {

if (other$modelFileName == null) {

break label602;

}

} else if (this$modelFileName.equals(other$modelFileName)) {

break label602;

}

return false;

}

label595: {

Object this$modelType = this.getModelType();

Object other$modelType = other.getModelType();

if (this$modelType == null) {

if (other$modelType == null) {

break label595;

}

} else if (this$modelType.equals(other$modelType)) {

break label595;

}

return false;

}

Object this$modelSize = this.getModelSize();

Object other$modelSize = other.getModelSize();

if (this$modelSize == null) {

if (other$modelSize != null) {

return false;

}

} else if (!this$modelSize.equals(other$modelSize)) {

return false;

}

Object this$modelSuffix = this.getModelSuffix();

Object other$modelSuffix = other.getModelSuffix();

if (this$modelSuffix == null) {

if (other$modelSuffix != null) {

return false;

}

} else if (!this$modelSuffix.equals(other$modelSuffix)) {

return false;

}

label574: {

Object this$modelLevel = this.getModelLevel();

Object other$modelLevel = other.getModelLevel();

if (this$modelLevel == null) {

if (other$modelLevel == null) {

break label574;

}

} else if (this$modelLevel.equals(other$modelLevel)) {

break label574;

}

return false;

}

label567: {

Object this$modelPath = this.getModelPath();

Object other$modelPath = other.getModelPath();

if (this$modelPath == null) {

if (other$modelPath == null) {

break label567;

}

} else if (this$modelPath.equals(other$modelPath)) {

break label567;

}

return false;

}

Object this$modelDesc = this.getModelDesc();

Object other$modelDesc = other.getModelDesc();

if (this$modelDesc == null) {

if (other$modelDesc != null) {

return false;

}

} else if (!this$modelDesc.equals(other$modelDesc)) {

return false;

}

label553: {

Object this$sysCode = this.getSysCode();

Object other$sysCode = other.getSysCode();

if (this$sysCode == null) {

if (other$sysCode == null) {

break label553;

}

} else if (this$sysCode.equals(other$sysCode)) {

break label553;

}

return false;

}

Object this$creator = this.getCreator();

Object other$creator = other.getCreator();

if (this$creator == null) {

if (other$creator != null) {

return false;

}

} else if (!this$creator.equals(other$creator)) {

return false;

}

label539: {

Object this$gmtCreate = this.getGmtCreate();

Object other$gmtCreate = other.getGmtCreate();

if (this$gmtCreate == null) {

if (other$gmtCreate == null) {

break label539;

}

} else if (this$gmtCreate.equals(other$gmtCreate)) {

break label539;

}

return false;

}

Object this$modifier = this.getModifier();

Object other$modifier = other.getModifier();

if (this$modifier == null) {

if (other$modifier != null) {

return false;

}

} else if (!this$modifier.equals(other$modifier)) {

return false;

}

Object this$gmtModified = this.getGmtModified();

Object other$gmtModified = other.getGmtModified();

if (this$gmtModified == null) {

if (other$gmtModified != null) {

return false;

}

} else if (!this$gmtModified.equals(other$gmtModified)) {

return false;

}

label518: {

Object this$isDeleted = this.getIsDeleted();

Object other$isDeleted = other.getIsDeleted();

if (this$isDeleted == null) {

if (other$isDeleted == null) {

break label518;

}

} else if (this$isDeleted.equals(other$isDeleted)) {

break label518;

}

return false;

}

label511: {

Object this$modelOrdering = this.getModelOrdering();

Object other$modelOrdering = other.getModelOrdering();

if (this$modelOrdering == null) {

if (other$modelOrdering == null) {

break label511;

}

} else if (this$modelOrdering.equals(other$modelOrdering)) {

break label511;

}

return false;

}

Object this$fileKey = this.getFileKey();

Object other$fileKey = other.getFileKey();

if (this$fileKey == null) {

if (other$fileKey != null) {

return false;

}

} else if (!this$fileKey.equals(other$fileKey)) {

return false;

}

Object this$fileKeys = this.getFileKeys();

Object other$fileKeys = other.getFileKeys();

if (this$fileKeys == null) {

if (other$fileKeys != null) {

return false;

}

} else if (!this$fileKeys.equals(other$fileKeys)) {

return false;

}

label490: {

Object this$businessIds = this.getBusinessIds();

Object other$businessIds = other.getBusinessIds();

if (this$businessIds == null) {

if (other$businessIds == null) {

break label490;

}

} else if (this$businessIds.equals(other$businessIds)) {

break label490;

}

return false;

}

label483: {

Object this$att4 = this.getAtt4();

Object other$att4 = other.getAtt4();

if (this$att4 == null) {

if (other$att4 == null) {

break label483;

}

} else if (this$att4.equals(other$att4)) {

break label483;

}

return false;

}

Object this$att5 = this.getAtt5();

Object other$att5 = other.getAtt5();

if (this$att5 == null) {

if (other$att5 != null) {

return false;

}

} else if (!this$att5.equals(other$att5)) {

return false;

}

Object this$att6 = this.getAtt6();

Object other$att6 = other.getAtt6();

if (this$att6 == null) {

if (other$att6 != null) {

return false;

}

} else if (!this$att6.equals(other$att6)) {

return false;

}

label462: {

Object this$dateAtt1 = this.getDateAtt1();

Object other$dateAtt1 = other.getDateAtt1();

if (this$dateAtt1 == null) {

if (other$dateAtt1 == null) {

break label462;

}

} else if (this$dateAtt1.equals(other$dateAtt1)) {

break label462;

}

return false;

}

label455: {

Object this$dateAtt2 = this.getDateAtt2();

Object other$dateAtt2 = other.getDateAtt2();

if (this$dateAtt2 == null) {

if (other$dateAtt2 == null) {

break label455;

}

} else if (this$dateAtt2.equals(other$dateAtt2)) {

break label455;

}

return false;

}

Object this$businessId = this.getBusinessId();

Object other$businessId = other.getBusinessId();

if (this$businessId == null) {

if (other$businessId != null) {

return false;

}

} else if (!this$businessId.equals(other$businessId)) {

return false;

}

label441: {

Object this$numAtt2 = this.getNumAtt2();

Object other$numAtt2 = other.getNumAtt2();

if (this$numAtt2 == null) {

if (other$numAtt2 == null) {

break label441;

}

} else if (this$numAtt2.equals(other$numAtt2)) {

break label441;

}

return false;

}

Object this$numAtt3 = this.getNumAtt3();

Object other$numAtt3 = other.getNumAtt3();

if (this$numAtt3 == null) {

if (other$numAtt3 != null) {

return false;

}

} else if (!this$numAtt3.equals(other$numAtt3)) {

return false;

}

label427: {

Object this$numAtt4 = this.getNumAtt4();

Object other$numAtt4 = other.getNumAtt4();

if (this$numAtt4 == null) {

if (other$numAtt4 == null) {

break label427;

}

} else if (this$numAtt4.equals(other$numAtt4)) {

break label427;

}

return false;

}

Object this$remark = this.getRemark();

Object other$remark = other.getRemark();

if (this$remark == null) {

if (other$remark != null) {

return false;

}

} else if (!this$remark.equals(other$remark)) {

return false;

}

Object this$length = this.getLength();

Object other$length = other.getLength();

if (this$length == null) {

if (other$length != null) {

return false;

}

} else if (!this$length.equals(other$length)) {

return false;

}

label406: {

Object this$height = this.getHeight();

Object other$height = other.getHeight();

if (this$height == null) {

if (other$height == null) {

break label406;

}

} else if (this$height.equals(other$height)) {

break label406;

}

return false;

}

label399: {

Object this$width = this.getWidth();

Object other$width = other.getWidth();

if (this$width == null) {

if (other$width == null) {

break label399;

}

} else if (this$width.equals(other$width)) {

break label399;

}

return false;

}

Object this$minHeight = this.getMinHeight();

Object other$minHeight = other.getMinHeight();

if (this$minHeight == null) {

if (other$minHeight != null) {

return false;

}

} else if (!this$minHeight.equals(other$minHeight)) {

return false;

}

Object this$isModelShare = this.getIsModelShare();

Object other$isModelShare = other.getIsModelShare();

if (this$isModelShare == null) {

if (other$isModelShare != null) {

return false;

}

} else if (!this$isModelShare.equals(other$isModelShare)) {

return false;

}

label378: {

Object this$isDecompress = this.getIsDecompress();

Object other$isDecompress = other.getIsDecompress();

if (this$isDecompress == null) {

if (other$isDecompress == null) {

break label378;

}

} else if (this$isDecompress.equals(other$isDecompress)) {

break label378;

}

return false;

}

label371: {

Object this$thumbPicPath = this.getThumbPicPath();

Object other$thumbPicPath = other.getThumbPicPath();

if (this$thumbPicPath == null) {

if (other$thumbPicPath == null) {

break label371;

}

} else if (this$thumbPicPath.equals(other$thumbPicPath)) {

break label371;

}

return false;

}

Object this$categoryIds = this.getCategoryIds();

Object other$categoryIds = other.getCategoryIds();

if (this$categoryIds == null) {

if (other$categoryIds != null) {

return false;

}

} else if (!this$categoryIds.equals(other$categoryIds)) {

return false;

}

Object this$categoryNames = this.getCategoryNames();

Object other$categoryNames = other.getCategoryNames();

if (this$categoryNames == null) {

if (other$categoryNames != null) {

return false;

}

} else if (!this$categoryNames.equals(other$categoryNames)) {

return false;

}

label350: {

Object this$transStatus = this.getTransStatus();

Object other$transStatus = other.getTransStatus();

if (this$transStatus == null) {

if (other$transStatus == null) {

break label350;

}

} else if (this$transStatus.equals(other$transStatus)) {

break label350;

}

return false;

}

label343: {

Object this$modelModelNum = this.getModelModelNum();

Object other$modelModelNum = other.getModelModelNum();

if (this$modelModelNum == null) {

if (other$modelModelNum == null) {

break label343;

}

} else if (this$modelModelNum.equals(other$modelModelNum)) {

break label343;

}

return false;

}

Object this$companyId = this.getCompanyId();

Object other$companyId = other.getCompanyId();

if (this$companyId == null) {

if (other$companyId != null) {

return false;

}

} else if (!this$companyId.equals(other$companyId)) {

return false;

}

label329: {

Object this$file3duPath = this.getFile3duPath();

Object other$file3duPath = other.getFile3duPath();

if (this$file3duPath == null) {

if (other$file3duPath == null) {

break label329;

}

} else if (this$file3duPath.equals(other$file3duPath)) {

break label329;

}

return false;

}

Object this$productId = this.getProductId();

Object other$productId = other.getProductId();

if (this$productId == null) {

if (other$productId != null) {

return false;

}

} else if (!this$productId.equals(other$productId)) {

return false;

}

label315: {

Object this$originId = this.getOriginId();

Object other$originId = other.getOriginId();

if (this$originId == null) {

if (other$originId == null) {

break label315;

}

} else if (this$originId.equals(other$originId)) {

break label315;

}

return false;

}

Object this$typeMark = this.getTypeMark();

Object other$typeMark = other.getTypeMark();

if (this$typeMark == null) {

if (other$typeMark != null) {

return false;

}

} else if (!this$typeMark.equals(other$typeMark)) {

return false;

}

Object this$smallTypeMark = this.getSmallTypeMark();

Object other$smallTypeMark = other.getSmallTypeMark();

if (this$smallTypeMark == null) {

if (other$smallTypeMark != null) {

return false;

}

} else if (!this$smallTypeMark.equals(other$smallTypeMark)) {

return false;

}

label294: {

Object this$typeNames = this.getTypeNames();

Object other$typeNames = other.getTypeNames();

if (this$typeNames == null) {

if (other$typeNames == null) {

break label294;

}

} else if (this$typeNames.equals(other$typeNames)) {

break label294;

}

return false;

}

label287: {

Object this$modelTextureInfo = this.getModelTextureInfo();

Object other$modelTextureInfo = other.getModelTextureInfo();

if (this$modelTextureInfo == null) {

if (other$modelTextureInfo == null) {

break label287;

}

} else if (this$modelTextureInfo.equals(other$modelTextureInfo)) {

break label287;

}

return false;

}

Object this$textureCount = this.getTextureCount();

Object other$textureCount = other.getTextureCount();

if (this$textureCount == null) {

if (other$textureCount != null) {

return false;

}

} else if (!this$textureCount.equals(other$textureCount)) {

return false;

}

Object this$mainProductFlag = this.getMainProductFlag();

Object other$mainProductFlag = other.getMainProductFlag();

if (this$mainProductFlag == null) {

if (other$mainProductFlag != null) {

return false;

}

} else if (!this$mainProductFlag.equals(other$mainProductFlag)) {

return false;

}

return true;

}

}

}

protected boolean canEqual(Object other) {

return other instanceof ModelBO;

}

public int hashCode() {

int PRIME = true;

int result = 1;

Object $id = this.getId();

int result = result \* 59 + ($id == null ? 43 : $id.hashCode());

Object $modelCode = this.getModelCode();

result = result \* 59 + ($modelCode == null ? 43 : $modelCode.hashCode());

Object $modelName = this.getModelName();

result = result \* 59 + ($modelName == null ? 43 : $modelName.hashCode());

Object $modelFileName = this.getModelFileName();

result = result \* 59 + ($modelFileName == null ? 43 : $modelFileName.hashCode());

Object $modelType = this.getModelType();

result = result \* 59 + ($modelType == null ? 43 : $modelType.hashCode());

Object $modelSize = this.getModelSize();

result = result \* 59 + ($modelSize == null ? 43 : $modelSize.hashCode());

Object $modelSuffix = this.getModelSuffix();

result = result \* 59 + ($modelSuffix == null ? 43 : $modelSuffix.hashCode());

Object $modelLevel = this.getModelLevel();

result = result \* 59 + ($modelLevel == null ? 43 : $modelLevel.hashCode());

Object $modelPath = this.getModelPath();

result = result \* 59 + ($modelPath == null ? 43 : $modelPath.hashCode());

Object $modelDesc = this.getModelDesc();

result = result \* 59 + ($modelDesc == null ? 43 : $modelDesc.hashCode());

Object $sysCode = this.getSysCode();

result = result \* 59 + ($sysCode == null ? 43 : $sysCode.hashCode());

Object $creator = this.getCreator();

result = result \* 59 + ($creator == null ? 43 : $creator.hashCode());

Object $gmtCreate = this.getGmtCreate();

result = result \* 59 + ($gmtCreate == null ? 43 : $gmtCreate.hashCode());

Object $modifier = this.getModifier();

result = result \* 59 + ($modifier == null ? 43 : $modifier.hashCode());

Object $gmtModified = this.getGmtModified();

result = result \* 59 + ($gmtModified == null ? 43 : $gmtModified.hashCode());

Object $isDeleted = this.getIsDeleted();

result = result \* 59 + ($isDeleted == null ? 43 : $isDeleted.hashCode());

Object $modelOrdering = this.getModelOrdering();

result = result \* 59 + ($modelOrdering == null ? 43 : $modelOrdering.hashCode());

Object $fileKey = this.getFileKey();

result = result \* 59 + ($fileKey == null ? 43 : $fileKey.hashCode());

Object $fileKeys = this.getFileKeys();

result = result \* 59 + ($fileKeys == null ? 43 : $fileKeys.hashCode());

Object $businessIds = this.getBusinessIds();

result = result \* 59 + ($businessIds == null ? 43 : $businessIds.hashCode());

Object $att4 = this.getAtt4();

result = result \* 59 + ($att4 == null ? 43 : $att4.hashCode());

Object $att5 = this.getAtt5();

result = result \* 59 + ($att5 == null ? 43 : $att5.hashCode());

Object $att6 = this.getAtt6();

result = result \* 59 + ($att6 == null ? 43 : $att6.hashCode());

Object $dateAtt1 = this.getDateAtt1();

result = result \* 59 + ($dateAtt1 == null ? 43 : $dateAtt1.hashCode());

Object $dateAtt2 = this.getDateAtt2();

result = result \* 59 + ($dateAtt2 == null ? 43 : $dateAtt2.hashCode());

Object $businessId = this.getBusinessId();

result = result \* 59 + ($businessId == null ? 43 : $businessId.hashCode());

Object $numAtt2 = this.getNumAtt2();

result = result \* 59 + ($numAtt2 == null ? 43 : $numAtt2.hashCode());

Object $numAtt3 = this.getNumAtt3();

result = result \* 59 + ($numAtt3 == null ? 43 : $numAtt3.hashCode());

Object $numAtt4 = this.getNumAtt4();

result = result \* 59 + ($numAtt4 == null ? 43 : $numAtt4.hashCode());

Object $remark = this.getRemark();

result = result \* 59 + ($remark == null ? 43 : $remark.hashCode());

Object $length = this.getLength();

result = result \* 59 + ($length == null ? 43 : $length.hashCode());

Object $height = this.getHeight();

result = result \* 59 + ($height == null ? 43 : $height.hashCode());

Object $width = this.getWidth();

result = result \* 59 + ($width == null ? 43 : $width.hashCode());

Object $minHeight = this.getMinHeight();

result = result \* 59 + ($minHeight == null ? 43 : $minHeight.hashCode());

Object $isModelShare = this.getIsModelShare();

result = result \* 59 + ($isModelShare == null ? 43 : $isModelShare.hashCode());

Object $isDecompress = this.getIsDecompress();

result = result \* 59 + ($isDecompress == null ? 43 : $isDecompress.hashCode());

Object $thumbPicPath = this.getThumbPicPath();

result = result \* 59 + ($thumbPicPath == null ? 43 : $thumbPicPath.hashCode());

Object $categoryIds = this.getCategoryIds();

result = result \* 59 + ($categoryIds == null ? 43 : $categoryIds.hashCode());

Object $categoryNames = this.getCategoryNames();

result = result \* 59 + ($categoryNames == null ? 43 : $categoryNames.hashCode());

Object $transStatus = this.getTransStatus();

result = result \* 59 + ($transStatus == null ? 43 : $transStatus.hashCode());

Object $modelModelNum = this.getModelModelNum();

result = result \* 59 + ($modelModelNum == null ? 43 : $modelModelNum.hashCode());

Object $companyId = this.getCompanyId();

result = result \* 59 + ($companyId == null ? 43 : $companyId.hashCode());

Object $file3duPath = this.getFile3duPath();

result = result \* 59 + ($file3duPath == null ? 43 : $file3duPath.hashCode());

Object $productId = this.getProductId();

result = result \* 59 + ($productId == null ? 43 : $productId.hashCode());

Object $originId = this.getOriginId();

result = result \* 59 + ($originId == null ? 43 : $originId.hashCode());

Object $typeMark = this.getTypeMark();

result = result \* 59 + ($typeMark == null ? 43 : $typeMark.hashCode());

Object $smallTypeMark = this.getSmallTypeMark();

result = result \* 59 + ($smallTypeMark == null ? 43 : $smallTypeMark.hashCode());

Object $typeNames = this.getTypeNames();

result = result \* 59 + ($typeNames == null ? 43 : $typeNames.hashCode());

Object $modelTextureInfo = this.getModelTextureInfo();

result = result \* 59 + ($modelTextureInfo == null ? 43 : $modelTextureInfo.hashCode());

Object $textureCount = this.getTextureCount();

result = result \* 59 + ($textureCount == null ? 43 : $textureCount.hashCode());

Object $mainProductFlag = this.getMainProductFlag();

result = result \* 59 + ($mainProductFlag == null ? 43 : $mainProductFlag.hashCode());

return result;

}

public String toString() {

return "ModelBO(id=" + this.getId() + ", modelCode=" + this.getModelCode() + ", modelName=" + this.getModelName() + ", modelFileName=" + this.getModelFileName() + ", modelType=" + this.getModelType() + ", modelSize=" + this.getModelSize() + ", modelSuffix=" + this.getModelSuffix() + ", modelLevel=" + this.getModelLevel() + ", modelPath=" + this.getModelPath() + ", modelDesc=" + this.getModelDesc() + ", sysCode=" + this.getSysCode() + ", creator=" + this.getCreator() + ", gmtCreate=" + this.getGmtCreate() + ", modifier=" + this.getModifier() + ", gmtModified=" + this.getGmtModified() + ", isDeleted=" + this.getIsDeleted() + ", modelOrdering=" + this.getModelOrdering() + ", fileKey=" + this.getFileKey() + ", fileKeys=" + this.getFileKeys() + ", businessIds=" + this.getBusinessIds() + ", att4=" + this.getAtt4() + ", att5=" + this.getAtt5() + ", att6=" + this.getAtt6() + ", dateAtt1=" + this.getDateAtt1() + ", dateAtt2=" + this.getDateAtt2() + ", businessId=" + this.getBusinessId() + ", numAtt2=" + this.getNumAtt2() + ", numAtt3=" + this.getNumAtt3() + ", numAtt4=" + this.getNumAtt4() + ", remark=" + this.getRemark() + ", length=" + this.getLength() + ", height=" + this.getHeight() + ", width=" + this.getWidth() + ", minHeight=" + this.getMinHeight() + ", isModelShare=" + this.getIsModelShare() + ", isDecompress=" + this.getIsDecompress() + ", thumbPicPath=" + this.getThumbPicPath() + ", categoryIds=" + this.getCategoryIds() + ", categoryNames=" + this.getCategoryNames() + ", transStatus=" + this.getTransStatus() + ", modelModelNum=" + this.getModelModelNum() + ", companyId=" + this.getCompanyId() + ", file3duPath=" + this.getFile3duPath() + ", productId=" + this.getProductId() + ", originId=" + this.getOriginId() + ", typeMark=" + this.getTypeMark() + ", smallTypeMark=" + this.getSmallTypeMark() + ", typeNames=" + this.getTypeNames() + ", modelTextureInfo=" + this.getModelTextureInfo() + ", textureCount=" + this.getTextureCount() + ", mainProductFlag=" + this.getMainProductFlag() + ")";

}

}

//

// Source code recreated from a .class file by IntelliJ IDEA

// (powered by Fernflower decompiler)

//

package com.sandu.api.resmodel.model.bo;

import io.swagger.annotations.ApiModelProperty;

import java.io.Serializable;

import javax.validation.constraints.Min;

import org.hibernate.validator.constraints.Range;

public class ModelAreaTextureRelBO implements Serializable {

@ApiModelProperty("贴图区域ID")

@Min(

value = 1L,

message = "请输入正确的ID"

)

private Integer areaId;

@ApiModelProperty("贴图ID")

@Min(

value = 1L,

message = "请输入正确的ID"

)

private Integer textureId;

@ApiModelProperty("是否为默认贴图:1是,0否")

@Range(

min = 0L,

max = 1L,

message = "请检查默认贴图参数是否正确"

)

private Integer isDefault;

@ApiModelProperty(

value = "影响价格",

hidden = true

)

@Range(

max = 1000000L,

message = "请输入正确的价格"

)

private Integer affectPrice;

public ModelAreaTextureRelBO() {

}

public Integer getAreaId() {

return this.areaId;

}

public Integer getTextureId() {

return this.textureId;

}

public Integer getIsDefault() {

return this.isDefault;

}

public Integer getAffectPrice() {

return this.affectPrice;

}

public void setAreaId(Integer areaId) {

this.areaId = areaId;

}

public void setTextureId(Integer textureId) {

this.textureId = textureId;

}

public void setIsDefault(Integer isDefault) {

this.isDefault = isDefault;

}

public void setAffectPrice(Integer affectPrice) {

this.affectPrice = affectPrice;

}

public boolean equals(Object o) {

if (o == this) {

return true;

} else if (!(o instanceof ModelAreaTextureRelBO)) {

return false;

} else {

ModelAreaTextureRelBO other = (ModelAreaTextureRelBO)o;

if (!other.canEqual(this)) {

return false;

} else {

label59: {

Object this$areaId = this.getAreaId();

Object other$areaId = other.getAreaId();

if (this$areaId == null) {

if (other$areaId == null) {

break label59;

}

} else if (this$areaId.equals(other$areaId)) {

break label59;

}

return false;

}

Object this$textureId = this.getTextureId();

Object other$textureId = other.getTextureId();

if (this$textureId == null) {

if (other$textureId != null) {

return false;

}

} else if (!this$textureId.equals(other$textureId)) {

return false;

}

Object this$isDefault = this.getIsDefault();

Object other$isDefault = other.getIsDefault();

if (this$isDefault == null) {

if (other$isDefault != null) {

return false;

}

} else if (!this$isDefault.equals(other$isDefault)) {

return false;

}

Object this$affectPrice = this.getAffectPrice();

Object other$affectPrice = other.getAffectPrice();

if (this$affectPrice == null) {

if (other$affectPrice != null) {

return false;

}

} else if (!this$affectPrice.equals(other$affectPrice)) {

return false;

}

return true;

}

}

}

protected boolean canEqual(Object other) {

return other instanceof ModelAreaTextureRelBO;

}

public int hashCode() {

int PRIME = true;

int result = 1;

Object $areaId = this.getAreaId();

int result = result \* 59 + ($areaId == null ? 43 : $areaId.hashCode());

Object $textureId = this.getTextureId();

result = result \* 59 + ($textureId == null ? 43 : $textureId.hashCode());

Object $isDefault = this.getIsDefault();

result = result \* 59 + ($isDefault == null ? 43 : $isDefault.hashCode());

Object $affectPrice = this.getAffectPrice();

result = result \* 59 + ($affectPrice == null ? 43 : $affectPrice.hashCode());

return result;

}

public String toString() {

return "ModelAreaTextureRelBO(areaId=" + this.getAreaId() + ", textureId=" + this.getTextureId() + ", isDefault=" + this.getIsDefault() + ", affectPrice=" + this.getAffectPrice() + ")";

}

}

//

// Source code recreated from a .class file by IntelliJ IDEA

// (powered by Fernflower decompiler)

//

package com.sandu.api.resmodel.input;

import io.swagger.annotations.ApiModelProperty;

import java.beans.ConstructorProperties;

import java.io.Serializable;

import java.util.List;

import javax.validation.constraints.Min;

import javax.validation.constraints.NotNull;

import javax.validation.constraints.Pattern;

import org.hibernate.validator.constraints.Length;

import org.hibernate.validator.constraints.NotEmpty;

import org.hibernate.validator.constraints.Range;

public class ResModelAdd implements Serializable {

@ApiModelProperty(

value = "模型名称",

required = true

)

@NotEmpty(

message = "模型名称不能为空"

)

@Length(

max = 100,

message = "模型名称不能超过{max}个字"

)

private String modelName;

@ApiModelProperty("模型路径")

@Pattern(

regexp = ".+(\\.)((?i)(assetbundle)|(3du))$",

message = "请选择正确的文件格式"

)

@Length(

max = 200,

message = "模型路径长度不能超过{max}"

)

private String modelPath;

@ApiModelProperty("备注")

@Length(

max = 200,

message = "长度不能超过{max}"

)

private String modelDesc;

@ApiModelProperty("模型缩略图路径")

@Length(

max = 200,

message = "模型缩略图路径长度不能超过{max}"

)

private String thumbPicPath;

@ApiModelProperty("模型分类")

@Length(

max = 30,

message = "模型分类长度不能超过{max}"

)

private String categoryIds;

@ApiModelProperty("模型分类")

@Length(

max = 30,

message = "模型分类长度不能超过{max}"

)

private String categoryNames;

@ApiModelProperty("模型型号")

@Length(

max = 30,

message = "模型型号长度不能超过{max}"

)

private String modelModelNum;

@ApiModelProperty("模型转化状态")

@Length(

max = 30,

message = "模型转化状态长度不能超过{max}"

)

private String transStatus;

@ApiModelProperty("企业ID")

@NotNull(

message = "企业ID不能为空"

)

private Integer companyId;

@ApiModelProperty("用户ID")

@NotNull(

message = "用户ID不能为空"

)

@Min(

value = 1L,

message = "用户id不合法"

)

private Integer userId;

@ApiModelProperty("长")

@Range(

min = 0L,

max = 100000L,

message = "长度 最长 {max} 最短{min}"

)

private Integer length;

@ApiModelProperty("宽")

@Range(

min = 0L,

max = 100000L,

message = "宽度 最长 {max} 最短{min}"

)

private Integer width;

@ApiModelProperty("高")

@Range(

min = 0L,

max = 100000L,

message = "高度 最长 {max} 最短{min}"

)

private Integer height;

@ApiModelProperty(

hidden = true

)

private Long originId;

@ApiModelProperty(

value = "模型FileKey",

hidden = true

)

private String fileKey;

@ApiModelProperty(

value = "材质对应材质球ID,仅在模型编辑器上传模型使用",

hidden = true

)

private Integer ball2TexturePicId;

@ApiModelProperty("区域编码")

private String areaCode;

@ApiModelProperty("模型小类编码")

private String smallType;

@ApiModelProperty("模型区域信息编码")

private List<ModelAreaTextureRelAdd> modelAreaTextureRelAdd;

@ApiModelProperty(

value = "硬装产品主模型标志(1:主模型,0:副模型)",

hidden = true

)

private Integer mainModelFlag;

public static ResModelAdd.ResModelAddBuilder builder() {

return new ResModelAdd.ResModelAddBuilder();

}

public String getModelName() {

return this.modelName;

}

public String getModelPath() {

return this.modelPath;

}

public String getModelDesc() {

return this.modelDesc;

}

public String getThumbPicPath() {

return this.thumbPicPath;

}

public String getCategoryIds() {

return this.categoryIds;

}

public String getCategoryNames() {

return this.categoryNames;

}

public String getModelModelNum() {

return this.modelModelNum;

}

public String getTransStatus() {

return this.transStatus;

}

public Integer getCompanyId() {

return this.companyId;

}

public Integer getUserId() {

return this.userId;

}

public Integer getLength() {

return this.length;

}

public Integer getWidth() {

return this.width;

}

public Integer getHeight() {

return this.height;

}

public Long getOriginId() {

return this.originId;

}

public String getFileKey() {

return this.fileKey;

}

public Integer getBall2TexturePicId() {

return this.ball2TexturePicId;

}

public String getAreaCode() {

return this.areaCode;

}

public String getSmallType() {

return this.smallType;

}

public List<ModelAreaTextureRelAdd> getModelAreaTextureRelAdd() {

return this.modelAreaTextureRelAdd;

}

public Integer getMainModelFlag() {

return this.mainModelFlag;

}

public void setModelName(String modelName) {

this.modelName = modelName;

}

public void setModelPath(String modelPath) {

this.modelPath = modelPath;

}

public void setModelDesc(String modelDesc) {

this.modelDesc = modelDesc;

}

public void setThumbPicPath(String thumbPicPath) {

this.thumbPicPath = thumbPicPath;

}

public void setCategoryIds(String categoryIds) {

this.categoryIds = categoryIds;

}

public void setCategoryNames(String categoryNames) {

this.categoryNames = categoryNames;

}

public void setModelModelNum(String modelModelNum) {

this.modelModelNum = modelModelNum;

}

public void setTransStatus(String transStatus) {

this.transStatus = transStatus;

}

public void setCompanyId(Integer companyId) {

this.companyId = companyId;

}

public void setUserId(Integer userId) {

this.userId = userId;

}

public void setLength(Integer length) {

this.length = length;

}

public void setWidth(Integer width) {

this.width = width;

}

public void setHeight(Integer height) {

this.height = height;

}

public void setOriginId(Long originId) {

this.originId = originId;

}

public void setFileKey(String fileKey) {

this.fileKey = fileKey;

}

public void setBall2TexturePicId(Integer ball2TexturePicId) {

this.ball2TexturePicId = ball2TexturePicId;

}

public void setAreaCode(String areaCode) {

this.areaCode = areaCode;

}

public void setSmallType(String smallType) {

this.smallType = smallType;

}

public void setModelAreaTextureRelAdd(List<ModelAreaTextureRelAdd> modelAreaTextureRelAdd) {

this.modelAreaTextureRelAdd = modelAreaTextureRelAdd;

}

public void setMainModelFlag(Integer mainModelFlag) {

this.mainModelFlag = mainModelFlag;

}

public boolean equals(Object o) {

if (o == this) {

return true;

} else if (!(o instanceof ResModelAdd)) {

return false;

} else {

ResModelAdd other = (ResModelAdd)o;

if (!other.canEqual(this)) {

return false;

} else {

label251: {

Object this$modelName = this.getModelName();

Object other$modelName = other.getModelName();

if (this$modelName == null) {

if (other$modelName == null) {

break label251;

}

} else if (this$modelName.equals(other$modelName)) {

break label251;

}

return false;

}

Object this$modelPath = this.getModelPath();

Object other$modelPath = other.getModelPath();

if (this$modelPath == null) {

if (other$modelPath != null) {

return false;

}

} else if (!this$modelPath.equals(other$modelPath)) {

return false;

}

Object this$modelDesc = this.getModelDesc();

Object other$modelDesc = other.getModelDesc();

if (this$modelDesc == null) {

if (other$modelDesc != null) {

return false;

}

} else if (!this$modelDesc.equals(other$modelDesc)) {

return false;

}

label230: {

Object this$thumbPicPath = this.getThumbPicPath();

Object other$thumbPicPath = other.getThumbPicPath();

if (this$thumbPicPath == null) {

if (other$thumbPicPath == null) {

break label230;

}

} else if (this$thumbPicPath.equals(other$thumbPicPath)) {

break label230;

}

return false;

}

label223: {

Object this$categoryIds = this.getCategoryIds();

Object other$categoryIds = other.getCategoryIds();

if (this$categoryIds == null) {

if (other$categoryIds == null) {

break label223;

}

} else if (this$categoryIds.equals(other$categoryIds)) {

break label223;

}

return false;

}

label216: {

Object this$categoryNames = this.getCategoryNames();

Object other$categoryNames = other.getCategoryNames();

if (this$categoryNames == null) {

if (other$categoryNames == null) {

break label216;

}

} else if (this$categoryNames.equals(other$categoryNames)) {

break label216;

}

return false;

}

Object this$modelModelNum = this.getModelModelNum();

Object other$modelModelNum = other.getModelModelNum();

if (this$modelModelNum == null) {

if (other$modelModelNum != null) {

return false;

}

} else if (!this$modelModelNum.equals(other$modelModelNum)) {

return false;

}

label202: {

Object this$transStatus = this.getTransStatus();

Object other$transStatus = other.getTransStatus();

if (this$transStatus == null) {

if (other$transStatus == null) {

break label202;

}

} else if (this$transStatus.equals(other$transStatus)) {

break label202;

}

return false;

}

Object this$companyId = this.getCompanyId();

Object other$companyId = other.getCompanyId();

if (this$companyId == null) {

if (other$companyId != null) {

return false;

}

} else if (!this$companyId.equals(other$companyId)) {

return false;

}

label188: {

Object this$userId = this.getUserId();

Object other$userId = other.getUserId();

if (this$userId == null) {

if (other$userId == null) {

break label188;

}

} else if (this$userId.equals(other$userId)) {

break label188;

}

return false;

}

Object this$length = this.getLength();

Object other$length = other.getLength();

if (this$length == null) {

if (other$length != null) {

return false;

}

} else if (!this$length.equals(other$length)) {

return false;

}

Object this$width = this.getWidth();

Object other$width = other.getWidth();

if (this$width == null) {

if (other$width != null) {

return false;

}

} else if (!this$width.equals(other$width)) {

return false;

}

label167: {

Object this$height = this.getHeight();

Object other$height = other.getHeight();

if (this$height == null) {

if (other$height == null) {

break label167;

}

} else if (this$height.equals(other$height)) {

break label167;

}

return false;

}

label160: {

Object this$originId = this.getOriginId();

Object other$originId = other.getOriginId();

if (this$originId == null) {

if (other$originId == null) {

break label160;

}

} else if (this$originId.equals(other$originId)) {

break label160;

}

return false;

}

Object this$fileKey = this.getFileKey();

Object other$fileKey = other.getFileKey();

if (this$fileKey == null) {

if (other$fileKey != null) {

return false;

}

} else if (!this$fileKey.equals(other$fileKey)) {

return false;

}

Object this$ball2TexturePicId = this.getBall2TexturePicId();

Object other$ball2TexturePicId = other.getBall2TexturePicId();

if (this$ball2TexturePicId == null) {

if (other$ball2TexturePicId != null) {

return false;

}

} else if (!this$ball2TexturePicId.equals(other$ball2TexturePicId)) {

return false;

}

label139: {

Object this$areaCode = this.getAreaCode();

Object other$areaCode = other.getAreaCode();

if (this$areaCode == null) {

if (other$areaCode == null) {

break label139;

}

} else if (this$areaCode.equals(other$areaCode)) {

break label139;

}

return false;

}

Object this$smallType = this.getSmallType();

Object other$smallType = other.getSmallType();

if (this$smallType == null) {

if (other$smallType != null) {

return false;

}

} else if (!this$smallType.equals(other$smallType)) {

return false;

}

Object this$modelAreaTextureRelAdd = this.getModelAreaTextureRelAdd();

Object other$modelAreaTextureRelAdd = other.getModelAreaTextureRelAdd();

if (this$modelAreaTextureRelAdd == null) {

if (other$modelAreaTextureRelAdd != null) {

return false;

}

} else if (!this$modelAreaTextureRelAdd.equals(other$modelAreaTextureRelAdd)) {

return false;

}

Object this$mainModelFlag = this.getMainModelFlag();

Object other$mainModelFlag = other.getMainModelFlag();

if (this$mainModelFlag == null) {

if (other$mainModelFlag != null) {

return false;

}

} else if (!this$mainModelFlag.equals(other$mainModelFlag)) {

return false;

}

return true;

}

}

}

protected boolean canEqual(Object other) {

return other instanceof ResModelAdd;

}

public int hashCode() {

int PRIME = true;

int result = 1;

Object $modelName = this.getModelName();

int result = result \* 59 + ($modelName == null ? 43 : $modelName.hashCode());

Object $modelPath = this.getModelPath();

result = result \* 59 + ($modelPath == null ? 43 : $modelPath.hashCode());

Object $modelDesc = this.getModelDesc();

result = result \* 59 + ($modelDesc == null ? 43 : $modelDesc.hashCode());

Object $thumbPicPath = this.getThumbPicPath();

result = result \* 59 + ($thumbPicPath == null ? 43 : $thumbPicPath.hashCode());

Object $categoryIds = this.getCategoryIds();

result = result \* 59 + ($categoryIds == null ? 43 : $categoryIds.hashCode());

Object $categoryNames = this.getCategoryNames();

result = result \* 59 + ($categoryNames == null ? 43 : $categoryNames.hashCode());

Object $modelModelNum = this.getModelModelNum();

result = result \* 59 + ($modelModelNum == null ? 43 : $modelModelNum.hashCode());

Object $transStatus = this.getTransStatus();

result = result \* 59 + ($transStatus == null ? 43 : $transStatus.hashCode());

Object $companyId = this.getCompanyId();

result = result \* 59 + ($companyId == null ? 43 : $companyId.hashCode());

Object $userId = this.getUserId();

result = result \* 59 + ($userId == null ? 43 : $userId.hashCode());

Object $length = this.getLength();

result = result \* 59 + ($length == null ? 43 : $length.hashCode());

Object $width = this.getWidth();

result = result \* 59 + ($width == null ? 43 : $width.hashCode());

Object $height = this.getHeight();

result = result \* 59 + ($height == null ? 43 : $height.hashCode());

Object $originId = this.getOriginId();

result = result \* 59 + ($originId == null ? 43 : $originId.hashCode());

Object $fileKey = this.getFileKey();

result = result \* 59 + ($fileKey == null ? 43 : $fileKey.hashCode());

Object $ball2TexturePicId = this.getBall2TexturePicId();

result = result \* 59 + ($ball2TexturePicId == null ? 43 : $ball2TexturePicId.hashCode());

Object $areaCode = this.getAreaCode();

result = result \* 59 + ($areaCode == null ? 43 : $areaCode.hashCode());

Object $smallType = this.getSmallType();

result = result \* 59 + ($smallType == null ? 43 : $smallType.hashCode());

Object $modelAreaTextureRelAdd = this.getModelAreaTextureRelAdd();

result = result \* 59 + ($modelAreaTextureRelAdd == null ? 43 : $modelAreaTextureRelAdd.hashCode());

Object $mainModelFlag = this.getMainModelFlag();

result = result \* 59 + ($mainModelFlag == null ? 43 : $mainModelFlag.hashCode());

return result;

}

public String toString() {

return "ResModelAdd(modelName=" + this.getModelName() + ", modelPath=" + this.getModelPath() + ", modelDesc=" + this.getModelDesc() + ", thumbPicPath=" + this.getThumbPicPath() + ", categoryIds=" + this.getCategoryIds() + ", categoryNames=" + this.getCategoryNames() + ", modelModelNum=" + this.getModelModelNum() + ", transStatus=" + this.getTransStatus() + ", companyId=" + this.getCompanyId() + ", userId=" + this.getUserId() + ", length=" + this.getLength() + ", width=" + this.getWidth() + ", height=" + this.getHeight() + ", originId=" + this.getOriginId() + ", fileKey=" + this.getFileKey() + ", ball2TexturePicId=" + this.getBall2TexturePicId() + ", areaCode=" + this.getAreaCode() + ", smallType=" + this.getSmallType() + ", modelAreaTextureRelAdd=" + this.getModelAreaTextureRelAdd() + ", mainModelFlag=" + this.getMainModelFlag() + ")";

}

@ConstructorProperties({"modelName", "modelPath", "modelDesc", "thumbPicPath", "categoryIds", "categoryNames", "modelModelNum", "transStatus", "companyId", "userId", "length", "width", "height", "originId", "fileKey", "ball2TexturePicId", "areaCode", "smallType", "modelAreaTextureRelAdd", "mainModelFlag"})

public ResModelAdd(String modelName, String modelPath, String modelDesc, String thumbPicPath, String categoryIds, String categoryNames, String modelModelNum, String transStatus, Integer companyId, Integer userId, Integer length, Integer width, Integer height, Long originId, String fileKey, Integer ball2TexturePicId, String areaCode, String smallType, List<ModelAreaTextureRelAdd> modelAreaTextureRelAdd, Integer mainModelFlag) {

this.modelName = modelName;

this.modelPath = modelPath;

this.modelDesc = modelDesc;

this.thumbPicPath = thumbPicPath;

this.categoryIds = categoryIds;

this.categoryNames = categoryNames;

this.modelModelNum = modelModelNum;

this.transStatus = transStatus;

this.companyId = companyId;

this.userId = userId;

this.length = length;

this.width = width;

this.height = height;

this.originId = originId;

this.fileKey = fileKey;

this.ball2TexturePicId = ball2TexturePicId;

this.areaCode = areaCode;

this.smallType = smallType;

this.modelAreaTextureRelAdd = modelAreaTextureRelAdd;

this.mainModelFlag = mainModelFlag;

}

public ResModelAdd() {

}

public static class ResModelAddBuilder {

private String modelName;

private String modelPath;

private String modelDesc;

private String thumbPicPath;

private String categoryIds;

private String categoryNames;

private String modelModelNum;

private String transStatus;

private Integer companyId;

private Integer userId;

private Integer length;

private Integer width;

private Integer height;

private Long originId;

private String fileKey;

private Integer ball2TexturePicId;

private String areaCode;

private String smallType;

private List<ModelAreaTextureRelAdd> modelAreaTextureRelAdd;

private Integer mainModelFlag;

ResModelAddBuilder() {

}

public ResModelAdd.ResModelAddBuilder modelName(String modelName) {

this.modelName = modelName;

return this;

}

public ResModelAdd.ResModelAddBuilder modelPath(String modelPath) {

this.modelPath = modelPath;

return this;

}

public ResModelAdd.ResModelAddBuilder modelDesc(String modelDesc) {

this.modelDesc = modelDesc;

return this;

}

public ResModelAdd.ResModelAddBuilder thumbPicPath(String thumbPicPath) {

this.thumbPicPath = thumbPicPath;

return this;

}

public ResModelAdd.ResModelAddBuilder categoryIds(String categoryIds) {

this.categoryIds = categoryIds;

return this;

}

public ResModelAdd.ResModelAddBuilder categoryNames(String categoryNames) {

this.categoryNames = categoryNames;

return this;

}

public ResModelAdd.ResModelAddBuilder modelModelNum(String modelModelNum) {

this.modelModelNum = modelModelNum;

return this;

}

public ResModelAdd.ResModelAddBuilder transStatus(String transStatus) {

this.transStatus = transStatus;

return this;

}

public ResModelAdd.ResModelAddBuilder companyId(Integer companyId) {

this.companyId = companyId;

return this;

}

public ResModelAdd.ResModelAddBuilder userId(Integer userId) {

this.userId = userId;

return this;

}

public ResModelAdd.ResModelAddBuilder length(Integer length) {

this.length = length;

return this;

}

public ResModelAdd.ResModelAddBuilder width(Integer width) {

this.width = width;

return this;

}

public ResModelAdd.ResModelAddBuilder height(Integer height) {

this.height = height;

return this;

}

public ResModelAdd.ResModelAddBuilder originId(Long originId) {

this.originId = originId;

return this;

}

public ResModelAdd.ResModelAddBuilder fileKey(String fileKey) {

this.fileKey = fileKey;

return this;

}

public ResModelAdd.ResModelAddBuilder ball2TexturePicId(Integer ball2TexturePicId) {

this.ball2TexturePicId = ball2TexturePicId;

return this;

}

public ResModelAdd.ResModelAddBuilder areaCode(String areaCode) {

this.areaCode = areaCode;

return this;

}

public ResModelAdd.ResModelAddBuilder smallType(String smallType) {

this.smallType = smallType;

return this;

}

public ResModelAdd.ResModelAddBuilder modelAreaTextureRelAdd(List<ModelAreaTextureRelAdd> modelAreaTextureRelAdd) {

this.modelAreaTextureRelAdd = modelAreaTextureRelAdd;

return this;

}

public ResModelAdd.ResModelAddBuilder mainModelFlag(Integer mainModelFlag) {

this.mainModelFlag = mainModelFlag;

return this;

}

public ResModelAdd build() {

return new ResModelAdd(this.modelName, this.modelPath, this.modelDesc, this.thumbPicPath, this.categoryIds, this.categoryNames, this.modelModelNum, this.transStatus, this.companyId, this.userId, this.length, this.width, this.height, this.originId, this.fileKey, this.ball2TexturePicId, this.areaCode, this.smallType, this.modelAreaTextureRelAdd, this.mainModelFlag);

}

public String toString() {

return "ResModelAdd.ResModelAddBuilder(modelName=" + this.modelName + ", modelPath=" + this.modelPath + ", modelDesc=" + this.modelDesc + ", thumbPicPath=" + this.thumbPicPath + ", categoryIds=" + this.categoryIds + ", categoryNames=" + this.categoryNames + ", modelModelNum=" + this.modelModelNum + ", transStatus=" + this.transStatus + ", companyId=" + this.companyId + ", userId=" + this.userId + ", length=" + this.length + ", width=" + this.width + ", height=" + this.height + ", originId=" + this.originId + ", fileKey=" + this.fileKey + ", ball2TexturePicId=" + this.ball2TexturePicId + ", areaCode=" + this.areaCode + ", smallType=" + this.smallType + ", modelAreaTextureRelAdd=" + this.modelAreaTextureRelAdd + ", mainModelFlag=" + this.mainModelFlag + ")";

}

}

}