#region 添加宝贝

public ApiResult Add(Children children)

{

ApiResult result = new ApiResult();

string message = string.Empty;

string ImagePath = DateTime.Now.ToString("yyyyMMddHHmmss") + ".jpg";

using (TransactionScope ts = new TransactionScope())

{

using (PreschoolEducationDBContext db = new PreschoolEducationDBContext())

{

try

{

SysUser loginUser = LoginHelper.CurrentUser(db);

IdWorker idWorker = new IdWorker(1);

if (!ClassUtil.Validate<Children>(children, ref message))

{

throw new ExceptionUtil(message);

}

if (!System.Enum.IsDefined(typeof(Sex), children.Sex))

{

throw new ExceptionUtil("请选择性别");

}

if (!System.Enum.IsDefined(typeof(ChildrenState), children.ChildrenState))

{

throw new ExceptionUtil("请选择在校状态");

}

if (!new BaseUtil().CheckIDCard(children.IDCard.Trim()))

{

throw new ExceptionUtil("身份证号错误,请重新输入");

}

if (db.Children.FirstOrDefault(t => t.IsDelete != true && t.ChildrenName == children.ChildrenName && t.FirstContactCellPhone == children.FirstContactCellPhone) != null)

{

throw new ExceptionUtil("已存在相同姓名和手机号的宝贝");

}

if (children.ChildrenState == ChildrenState.离园)

{

if (!System.Enum.IsDefined(typeof(LeaveReasonType), children.LeaveReasonType))

{

throw new ExceptionUtil("请选择离园原因");

}

if (!children.LeaveDate.HasValue)

{

throw new ExceptionUtil("请选择离园时间");

}

}

children.ID = idWorker.nextId();

children.ChildrenName = children.ChildrenName.Replace("/r/n", "").Trim();

children.IDCard = new BaseUtil().RSAEncrypt(children.IDCard.Trim());//加密处理

if (string.IsNullOrEmpty(children.Session))

{

children.Session = string.Empty;

}

children.SchoolID = loginUser.SchoolID;

children.CreateDate = DateTime.Now;

if (children.uploadList != null && children.uploadList.Count != 0)

{

if (children.uploadList.Count > 1)

{

throw new ExceptionUtil("儿童头像不超过1张");

}

children.ProfilePicture = children.uploadList[0].uploadUrl;

children.PictureKey = children.uploadList[0].osskey;

}

else

{

children.ProfilePicture = string.Empty;

children.PictureKey = string.Empty;

}

db.Children.Add(children);

db.SaveChanges();

//加入用户

//SysUser user = new SysUser();

//user.ID = idWorker.nextId();

//user.UserName = BaseUtil.GetSpellCode(children.ChildrenName);

//user.RealName = children.ChildrenName;

//user.Password = new BaseUtil().BuildPassword(BaseUtil.GetSpellCode(children.ChildrenName), "888888");

//user.CellPhone = children.FirstContactCellPhone;

//user.IDCard = children.IDCard;

//user.CreateDate = DateTime.Now;

//user.CreateUserID = loginUser.ID;

//user.IsAccountDisabled = false;

//user.IsDelete = false;

//user.SchoolID = loginUser.SchoolID;

//user.UserType = UserType.宝贝;

//user.RelationID = children.ID;

//user.ProfilePicture = children.ProfilePicture;

//db.SysUser.Add(user);

//db.SaveChanges();

new LogDAO().AddLog(LogCode.添加, "成功添加" + ClassUtil.GetEntityName(children) + ":" + children.ChildrenName, db);

result = ApiResult.NewSuccessJson(new JArray(), 0, "成功添加" + ClassUtil.GetEntityName(children) + children.ChildrenName);

ts.Complete();

}

catch (Exception ex)

{

message = ex.Message;

}

finally

{

ts.Dispose();

}

if (!string.IsNullOrEmpty(message))

{

result = ApiResult.NewErrorJson(LogCode.添加错误, message, db);

}

}

}

return result;

}

#endregion

#region 编辑宝贝

public ApiResult Edit(Children children)

{

ApiResult result = new ApiResult();

string message = string.Empty;

string ImagePath = DateTime.Now.ToString("yyyyMMddHHmmss") + ".jpg";

using (TransactionScope ts = new TransactionScope())

{

using (PreschoolEducationDBContext db = new PreschoolEducationDBContext())

{

try

{

Children oldChildren = db.Children.FirstOrDefault(t => t.IsDelete != true && t.ID == children.ID);

if (oldChildren == null)

{

throw new ExceptionUtil("原宝贝信息不存在");

}

if (!ClassUtil.Validate<Children>(children, ref message))

{

throw new ExceptionUtil(message);

}

if (!System.Enum.IsDefined(typeof(Sex), children.Sex))

{

throw new ExceptionUtil("请选择性别");

}

if (!System.Enum.IsDefined(typeof(ChildrenState), children.ChildrenState))

{

throw new ExceptionUtil("请选择在校状态");

}

if (!new BaseUtil().CheckIDCard(children.IDCard.Trim()))

{

throw new ExceptionUtil("身份证号错误,请重新输入");

}

if (db.Children.FirstOrDefault(t => t.IsDelete != true && t.ChildrenName == children.ChildrenName && t.FirstContactCellPhone == children.FirstContactCellPhone && t.ID != children.ID) != null)

{

throw new ExceptionUtil("已存在相同姓名和手机号的宝贝");

}

if (children.ChildrenState == ChildrenState.离园)

{

if (!System.Enum.IsDefined(typeof(LeaveReasonType), children.LeaveReasonType))

{

throw new ExceptionUtil("请选择离园原因");

}

if (!children.LeaveDate.HasValue)

{

throw new ExceptionUtil("请选择离园时间");

}

}

children.ChildrenName = children.ChildrenName.Replace("/r/n", "").Trim();

children.IDCard = new BaseUtil().RSAEncrypt(children.IDCard.Trim());//加密处理

new ClassUtil().EditEntity(oldChildren, children);

if (string.IsNullOrEmpty(oldChildren.Session))

{

oldChildren.Session = string.Empty;

}

//string path = System.Web.HttpContext.Current.Server.MapPath(oldChildren.ProfilePicture);

//if (File.Exists(path))

//{

// File.Delete(path);

//}

if (children.uploadList != null && children.uploadList.Count != 0)

{

if (children.uploadList.Count > 1)

{

throw new ExceptionUtil("宝贝头像不超过1张");

}

oldChildren.ProfilePicture = children.uploadList[0].uploadUrl;

oldChildren.PictureKey = children.uploadList[0].osskey;

}

else

{

oldChildren.ProfilePicture = string.Empty;

oldChildren.PictureKey = string.Empty;

}

db.SaveChanges();

//SysUser user = db.SysUser.FirstOrDefault(t => t.RelationID == oldChildren.ID);

//if (user != null)

//{

// user.RealName = oldChildren.ChildrenName;

// user.UserName = BaseUtil.GetSpellCode(oldChildren.ChildrenName);

// user.Password = new BaseUtil().BuildPassword(BaseUtil.GetSpellCode(oldChildren.ChildrenName), "888888");

// user.IDCard = oldChildren.IDCard;

// user.ProfilePicture = oldChildren.ProfilePicture;

// user.CellPhone = oldChildren.FirstContactCellPhone;

// db.SaveChanges();

//}

new LogDAO().AddLog(LogCode.添加, "成功修改" + ClassUtil.GetEntityName(children) + ":" + oldChildren.ChildrenName, db);

result = ApiResult.NewSuccessJson(new JArray(), 0, "成功修改" + ClassUtil.GetEntityName(children) + oldChildren.ChildrenName);

ts.Complete();

}

catch (Exception ex)

{

message = ex.Message;

}

finally

{

ts.Dispose();

}

if (!string.IsNullOrEmpty(message))

{

result = ApiResult.NewErrorJson(LogCode.修改错误, message, db);

}

}

}

return result;

}

#endregion

#region 删除

public ApiResult Delete(List<long> IDList)

{

ApiResult result = new ApiResult();

string message = string.Empty;

using (PreschoolEducationDBContext db = new PreschoolEducationDBContext())

{

try

{

if (IDList.Count == 0 || IDList == null)

{

throw new ExceptionUtil("请选择要删除的宝贝信息");

}

List<Children> childrenList = db.Children.Where(t => IDList.Contains(t.ID)).ToList();

childrenList.ForEach(t => t.IsDelete = true);

db.SaveChanges();

new LogDAO().AddLog(LogCode.删除, "成功删除宝贝：" + string.Join(",", childrenList.Select(t => t.ChildrenName).ToArray()), db);

result = ApiResult.NewSuccessJson(new JArray(), 0, "成功删除宝贝");

}

catch (Exception ex)

{

message = ex.Message;

}

if (!string.IsNullOrEmpty(message))

{

result = ApiResult.NewErrorJson(LogCode.删除错误, message, db);

}

}

return result;

}

#endregion

#region 获取宝贝

public ApiResult Get(long id)

{

ApiResult result = new ApiResult();

string message = string.Empty;

using (PreschoolEducationDBContext db = new PreschoolEducationDBContext())

{

try

{

Children selectChildren = db.Children.FirstOrDefault(t => t.IsDelete != true && t.ID == id);

if (selectChildren == null)

{

throw new ExceptionUtil("该宝贝信息不存在");

}

List<AccessPost> uploadList = new List<AccessPost>();

uploadList.Add(new AccessPost

{

name = "",

url = string.IsNullOrEmpty(selectChildren.ProfilePicture) ? (selectChildren.Sex == Sex.男 ? SchoolURL + ParaUtil.ResourceImageHtmlPath + "男同学.png" : SchoolURL + ParaUtil.ResourceImageHtmlPath + "女同学.png") : selectChildren.ProfilePicture,

uploadUrl = selectChildren.ProfilePicture

});

result = ApiResult.NewSuccessJson(new

{

selectChildren.ID,

selectChildren.ChildrenName,

BirthDay = selectChildren.BirthDay.ToString("yyyy-MM-dd"),

selectChildren.Sex,

SexName = System.Enum.GetName(typeof(Sex), selectChildren.Sex),

JoinSchoolDate = selectChildren.JoinSchoolDate.ToString("yyyy-MM-dd"),

selectChildren.ChildrenState,

ChildrenStateName = System.Enum.GetName(typeof(ChildrenState), selectChildren.ChildrenState),

IDCard = new BaseUtil().RSADecrypt(selectChildren.IDCard).Replace("\0", ""),

selectChildren.Session,

ClassID = selectChildren.ClassID.ToString(),

ClassName = db.Class.FirstOrDefault(t => t.ID == selectChildren.ClassID) != null ? db.Class.FirstOrDefault(t => t.ID == selectChildren.ClassID).ClassName : "",

selectChildren.BloodType,

BloodTypeName = selectChildren.BloodType == null ? "" : System.Enum.GetName(typeof(BloodType), selectChildren.BloodType),

selectChildren.FirstContactName,

selectChildren.FirstContactCellPhone,

selectChildren.FirstRelationship,

selectChildren.SecondContactName,

selectChildren.SecondContactCellPhone,

selectChildren.SecondRelationship,

selectChildren.RegisteredResidence,

selectChildren.Address,

//selectChildren.Allergy,

ProfilePicture = string.IsNullOrEmpty(selectChildren.ProfilePicture) ? (selectChildren.Sex == Sex.男 ? SchoolURL + ParaUtil.ResourceImageHtmlPath + "男宝贝.png" : SchoolURL + ParaUtil.ResourceImageHtmlPath + "女宝贝.png") : selectChildren.ProfilePicture,

uploadList,

selectChildren.LeaveReasonType,

selectChildren.LeaveDesc,

LeaveDate = selectChildren.LeaveDate.HasValue?selectChildren.LeaveDate.Value.ToString("yyyy-MM-dd"):"",

LeaveReasonTypeName = selectChildren.ChildrenState == ChildrenState.离园 ? System.Enum.GetName(typeof(LeaveReasonType), selectChildren.LeaveReasonType):""

});

}

catch (Exception ex)

{

message = ex.Message;

}

if (!string.IsNullOrEmpty(message))

{

result = ApiResult.NewErrorJson(LogCode.获取错误, message, db);

}

}

return result;

}

#endregion

#region 宝贝列表

public ApiResult List(long? schoolID = null, string session = "", long classID = -2, string childrenName = "", int page = 1, int limit = 10)

{

ApiResult result = new ApiResult();

string message = string.Empty;

session = session ?? "";

childrenName = childrenName ?? "";

int year = DateTime.Now.Year;

using (PreschoolEducationDBContext db = new PreschoolEducationDBContext())

{

try

{

SysUser loginUser = LoginHelper.CurrentUser(db);

School currentSchool = LoginHelper.CurrentSchool(db);

List<School> schoolList = new BaseUtil().GetSchoolList(currentSchool, db, schoolID);

List<long> schoolIDList = schoolList.Select(t => t.ID).ToList();

IQueryable<Children> childrenIQueryable = db.Children.Where(t => schoolIDList.Contains(t.SchoolID) && t.IsDelete != true && t.ChildrenState == ChildrenState.在园 && t.ChildrenName.Contains(childrenName) && t.Session.Contains(session) && (classID == -2 || t.ClassID == classID)).OrderByDescending(t => t.CreateDate);

int total = childrenIQueryable.Count();

List<Children> childrenList = childrenIQueryable.Skip((page - 1) \* limit).Take(limit).ToList();

List<long?> classIDList = childrenList.Select(t => t.ClassID).Distinct().ToList();

List<Class> classList = db.Class.Where(t => classIDList.Contains(t.ID)).ToList();

List<object> returnList = new List<object>();

if (childrenList.Count != 0)

{

foreach (Children children in childrenList)

{

returnList.Add(new

{

children.ID,

children.ChildrenName,

Birthday = children.BirthDay.ToString("yyyy-MM-dd"),

SexName = System.Enum.GetName(typeof(Sex), children.Sex),

JoinSchoolDate = children.JoinSchoolDate.ToString("yyyy-MM-dd"),

children.Session,

ClassName = classList.FirstOrDefault(t => t.ID == children.ClassID) != null ? classList.FirstOrDefault(t => t.ID == children.ClassID).ClassName : "",

children.FirstContactName,

children.FirstContactCellPhone,

//children.Allergy,

SchoolName = schoolList.FirstOrDefault(t => t.ID == children.SchoolID).Name,

Age = year - children.BirthDay.Year + 1,

IsEdit = loginUser.SchoolID == children.SchoolID

});

}

result = ApiResult.NewSuccessJson(returnList, total);

}

else

{

result = ApiResult.NewSuccessJson(returnList, 0, "暂无数据");

}

}

catch (Exception ex)

{

message = ex.Message.ToString();

}

if (!string.IsNullOrEmpty(message))

{

result = ApiResult.NewErrorJson(LogCode.获取错误, message, db);

}

}

return result;

}

#endregion